RUBRIQUE POQUETTES

ISSN 0242-9799

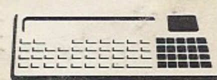
TRACE

TIREZ PLUS DE VOTRE TRS-80*

NUMERO 3





N° 3 - 3e trimestre 1982 Canada : 6.50 $ - Belgique : 200 FB - Suisse : 11 FS 25 F

TRACE n° 3

Sommaire

Editorial

L'année 1982 marque un tournant décisif de l'informatique individuelle : celui de la polarisation, de la structuration des produits et des politiques des entreprises. En effet, alors qu'arrivent les nouveaux systèmes 16 bits, tels celui d'IBM ou le Modèle 16, une autre poussée aussi forte s'exerce sur les systèmes à moins de 5000 Francs, tant du côté des ordinateurs de poche, que du côté des ordinateurs "de valise" comme les récents Epson HX-20 et Hewlett-Packard 75C, et enfin autant que du côté des "ordinateurs de famille" comme le TRS couleur ou le Thomson T-07 toujours attendu.

Les constructeurs se montrent du coup plutôt prudents quant à d'éventuels ordinateurs situés dans la tranche intermédiaire, et peu osent s'aventurer à y proposer de nouveaux systèmes. Tout au plus voit-on fleurir les copies améliorées des valeurs sûres que semblent être l'Apple 2 et le TRS-80 modèle 1 ou 3, ces matériels semblant eux continuer leur bonhomme de chemin, même si au passage le modèle 1 a été abandonné aux Etats-Unis : aucune importance, Tandy le soutient encore à bout de bras, lui adjoignant même récemment un complément qui lui permet d'utiliser les minidisquettes du modèle 3.

Du côté des entreprises elles-mêmes, c'est entre deux grands axes que la structuration s'effectue : la production et la distribution. A l'heure actuelle seul Tandy semble pouvoir avec quelque bonheur faire les deux simultanément. Mais il est vrai que cela ne va pas sans lâcher du lest : côté production, en achetant ses poquettes PC-1 et PC-2 à Sharp, et en passant des accords avec différents constructeurs orientaux et avec Matra. Côté distribution, en se tournant vers des ensembliers informatiques comme MTI, ou vers des réseaux complémentaires du sien, tel celui de RCA pour diffuser le "couleur", ou en France American Express pour le modèle 3. Bref, Tandy se défend pour préserver ses deux activités sans devoir sacrifier l'une à l'autre. Mais il n'est pas évident que le créneau du haut de gamme ne fera pas, après celui du milieu, les frais de l'opération : tout va se jouer sur le modèle 16. Aura-t-il dans deux ans suffisamment de logiciels pout paraître plus qu'une machine techniquement séduisante?

TRACE

Rédacteur en chef : Alain Pinaud - *Editeur :* Bernard Savonet - *Directeur de la publication :* Jean-Pierre Nizard - *Rédaction et abonnements :* Editrace, 8 rue Saint-Marc, 75002 Paris. - *Régie publicitaire :* Force 7, 41 rue de la Grange-aux-Belles, 75483 Paris Cedex 10 - Tél. (1) 238 66 10. - *Diffusion auprès des boutiques informatiques et libraires :* Editions du PSI, 41 rue Jacquard - BP 86 - 77400 Lagny - Tél. (6) 007 59 31.

Abonnements 4 numéros (un an) : France 85 FF ; Belgique 680 FB ; Suisse 37 FS ; Canada 22 $C ; Etranger 100 FF.

Ont collaboré à ce numéro : Kaarina Alain, Nicole Aleman, Tristan d'Amico, Jean-David Blanc, Christian Blanchet, Jean-Claude Bouman, Ivan Boussion, Pierre Bréhon, Roger Buvat, Sylvine Dautref, Alain Gandossi, Cyril Grandpierre, Pierre Landry, Wolfgang Lauter, Alain Marmoiton, Frédéric Mora, Claude Nowakowski, Rémy Pineau, Mireille Piquet, Jean-Paul du Tillet, Christophe Yonnet.

SIVEA SIVEA SI

COMPATIBLE LOGICIELS TRS 80*

PROMOTION!

CLAVIER / UNITÉ CENTRALE / MAGNÉTOPHONE :
EG 3003 - .... **3800 F TTC**

- Microprocesseur Z 80 • 16 K RAM utilisateur
- ROM BASIC Microsoft LEVEL 2 (12 K)
- Affichage écran : 16 lignes de 64 ou 32 caractères
- Graphismes : 128 x 48 points • Minuscules à l'écran
- Branchement direct sur moniteur vidéo
- Magnétophone à cassettes intégré au boîtier
- Vu-mètre de réglage de niveau de lecture magnétophone
- Prise DIN pour un second magnétophone
- Fonction RENUMBER
- Alimentation intégrée 200/240 volts 50 Hz
- Sortie son avec haut-parleur incorporé
- Livré avec manuels en français, cordons et une cassette de démonstration.
- **Compatible logiciels TRS 80***



INTERFACE D'EXTENSION AVEC 32 K RAM - .... : **3500 F TTC**
Boîtier d'extention contenant le contrôleur de floppy-disks, interface parallèle Centronics, interface Bus S100, 32 K RAM (qui étendent donc votre système à un total de 48 K RAM). Fourni avec alimentation secteur, manuel et cordon de raccordement au clavier.

DRIVE SIMPLE
(Floppy-Disk Drive) - .... **3995 F TTC**
Lecteur enregistreur de disques souples 5 pouces 1/4 - 40 pistes - 100 K. Totalement compatible pour la lecture de logiciels TRS 80* sur disque.

ENSEMBLE DE 2 DRIVES .... **7700 F TTC**

LANGAGES

- **Editeur-Assembleur MICROSOFT**
  K7, 16 K - .... **350 F TTC**
  Disk, 48 K - .... **395 F TTC**
- **Infinite Basic**
  K7, 16 K - .... **495 F TTC**



EXCEPTIONNEL!

IMPRIMANTES

- SEIKOSHA GP 100 A .... **2295 F TTC**
- OKI MICROLINE 80 .... **2995 F TTC**

Offre valable dans la limite des stocks disponibles.

JEUX POUR TRS-80 ET VIDEO-GENIE

- **DAMES CHALLENGER K7**
  Dames françaises contre l'ordinateur, partenaire de très haut niveau K7, 16K .... **220 F TTC**
- **T 80 FLIGHT SIMULATOR K7**
  Simulateur de pilotage d'avion. Graphismes 3 DIM en temps réel K7, 16 K .... **270 F TTC**
- **LES ENVAHISSEURS**
  Animation graphique très rapide. Détruisez les hordes d'envahisseurs qui foncent vers la terre K7, 16 K .... **110 F TTC**
- **TANKTICS**
  Superbe wargame contre l'ordinateur. Simulation de batailles de chars durant la seconde guerre mondiale K7, 16 K .... **295 F TTC**
- **SPACE SHUTTLE**
  Pilotez la navette spatiale américaine K7, 16 K ...... **170 F TTC**
- **ASYLUM**
  Aventure avec graphismes 3 DIM : Evadez-vous de l'hopital psychiatrique où vous avez été interné par erreur! K7, 16 K .... **211 F TTC**
- **DEFENSE COMMAND**
  Jeu d'adresse avec superbe animation graphique : défendez vos réserves de carburant contre des hordes d'ennemis d'une autre galaxie K7, 16K .... **160 F TTC**
- **ROBOT ATTACK**
  Jeu d'adresse avec animation graphique, sonore et parlante! Détruisez tous les robots qui ont envahi la station spatiale K7, 16 K .... **160 F TTC**
- **STELLAR ESCORT**
  Un nouveau jeu BIG FIVE : animation graphique rapide, effets sonores. Détruisez les vaisseaux ennemis qui apparaissent dans le viseur de vos canons laser! K7, 16 K .... **165 F TTC**

*** TRS 80 est un modèle déposé par TANDY-RADIO-SHACK.**

LE NOUVEAU CATALOGUE SIVEA VIENT DE PARAITRE !
Pour le recevoir GRATUITEMENT retournez nous le Bon de Cde avec seulement vos nom et adresse.

VEA

31, Bd des Batignolles
75008 PARIS
Tél. : 522.70.66 (+) – Télex 280 902 F
Métro : Rome - Place Clichy
parking assuré au 43 bis, Bd des Batignolles

21 bis, rue de Valmy
59000 LILLE
Tél. : (20) 57.88.43 (+)
Fermé le lundi matin jusqu'à 13 h 30
parking assuré Bd de la Liberté

OUVERT SANS INTERRUPTION DU LUNDI AU SAMEDI DE 9 H 30 A 18 H 30 (LUNDI 13 H 30 POUR LILLE).
DETAXE A L'EXPORTATION. VENTE PAR CORRESPONDANCE – CREDIT – LEASING – CARTE BLEUE-VISA



SIVEA

LIBRAIRIE POUR TRS 80* ET VIDEO-GENIE

- **THE BOOK Vol. 1 :**
Toutes les routines arithmétiques de la ROM pour programmer plus facilement en langage machine. 120 pages .... **N.C.**
- **THE BOOK Vol. 2 :**
Les routines d'entrée/sortie de la ROM : écran, clavier, cassettes. 140 pages .... **135 F TTC**
- **LE MANUEL D'UTILISATION DU DOS :**
Concerne TRS DOS (Tandy) et New-Dos Plus (Apparat) : les commandes, les utilitaires, etc. 90 pages en français .... **95 F TTC**
- **DISK INTERFACING GUIDE :**
La programmation des formatages, lectures et écritures sur disquette en langage machine : accès disque sans utiliser le DOS. Liste des commandes du contrôleur de disques, etc. 55 pages .... **70 F TTC**
- **TRS 80 DISK & OTHER MYSTERIES :**
Tout sur les disquettes du TRS 80 : architecture du directory, organisation des fichiers, comment récupérer des données ou des programmes sur disquette abîmée, etc. 130 pages .... **195 F TTC**
- **MICROSOFT BASIC DECODED & OTHER MYSTERIES :**
Toutes les routines de la ROM utilisables pour programmer en langage machine : adresse, contenu des registres, etc. Organisation et déroulement d'un programme Basic : où il commence, où il stocke ses données, etc.
310 pages .... **260 F TTC**





- **BASIC FASTER AND BETTER & OTHER MYSTERIES :**
Enorme recueil de "trucs" de programmation de haut niveau : string packing, line packing, etc. 290 pages .... **290 F TTC**



- **THE CUSTOM TRS & OTHER MYSTERIES :**
Tous sur le hardware du TRS-80. Nombreux montages pour améliorer votre TRS : haute résolution graphique, inversion vidéo, majuscules/minuscules, comment faire de la musique, etc.
330 pages .... **295 F TTC**

- **TRS 80 GRAPHICS FOR MODEL 1 and MODEL 3 (Byte Books) :**
Très bel ouvrage abondament illustré sur le graphisme TRS-80 / SET et RESET, POKE, langage machine, etc. 290 pages .... **175 F TTC**
- **TRS-80 ASSEMBLY LANGUAGE (Prentice-Hall) :**
La programmation en langage machine sur TRS 80.
190 pages .... **140 F TTC**
- **TRS 80 Interfacing Book 1 :**
Comment faire communiquer le TRS-80 avec son environnement : convertisseur D/A, etc. 190 pages .... **105 F TTC**
- **TRS 80 Interfacing Book 2 :**
Encore d'autres circuits d'interface. 250 pages .... **105 F TTC**
- **32 BASIC Programs for TRS-80** .... **110 F TTC**
- **TRS-80 Assembly language made simple** .... **225 F TTC**
- **THE SOFTSIDE SAMPLER,** 29 programmes de jeux ...... **130 F TTC**
- **LA PRATIQUE DU TRS 80 BASIC 2 (H. LILEN) :** ..... **70 F TTC**
- **LA PRATIQUE DU TRS 80 Vol. 1 (Editions P.S.I.) :** ... **65 F TTC**
- **LA PRATIQUE DU TRS 80 Vol. 2 (Editions P.S.I.) :** ... **85 F TTC**
- **LA PRATIQUE DU TRS 80 Vol. 3 (Editions P.S.I.) :** ... **75 F TTC**
- **LES GRAPHIQUES SUR TRS 80 (Editions P.S.I.) :** ..... **75 F TTC**
- **LES GRAPHIQUES SUR TRS-80 :**
Tout sur les possibilités graphiques du TRS-80.
140 pages en français .... **75 F TTC**
- **THE POWER OF VISICALC Vol. 1 :**
Découvrez toutes les fantastiques possibilités d'application de Visicalc dans l'entreprise et au foyer : comptabilité, finances, fiscalité, etc.
88 pages .... **115 F TTC**
- **THE POWER OF VISICALC Vol. 2 :**
Etude approfondie de Visicalc : l'instruction IF, etc.
97 pages .... **115 F TTC**

TR 9

BON DE COMMANDE

Retourner à : SIVEA S.A. - 31, Bd des Batignolles 75008 PARIS

Nom Prénom
Adresse

Ville
Code Postal Bureau Distributeur
Pays Téléphone

Quantité	DESIGNATION	PRIX UNIT	PRIX TOTAL

MODE DE REGLEMENT :	TOTAL	
Cheque bancaire joint ☐	Participation port et emballage : Voir aussi ci-dessous : + 220 F	+ **30** F
CCP joint ☐	Contre-Remboursement : + 30 F (France seulement)	
Mandat-lettre joint ☐	Etranger et DOM-TOM : + 30 F	
Contre-Remboursement ☐	TOTAL	

FRAIS DE PORT ET D'EMBALLAGE
Ajouter **220 F** pour toute commande comportant, pour tout ou partie, des articles tels que : unité centrale, floppy-disk, imprimante, moniteur vidéo, papier pour imprimante. Transport par service express avec assurance comprise (Corse, DOM-TOM étranger nous consulter par téléphone ou courrier en ce cas)

NEW ! **SUPERSOFT** pour NEW !

TRS-80*

GRAF transforme votre imprimante LINEPRINTER VI en table à dessin de 65536 points. GRAF fonctionne en ajoutant des commandes au BASIC DISK. Simple à utiliser, performant. Utile pour tracer des courbes, faire du graphisme.
MOD I DISK ...... **309 F**

GESTCO gère un compte bancaire personnel 1000 à 2000 Mvts. Calcul du solde. Liste écran, imprimante. TRI sur code ou libellé.
MOD I DISK ...... **290 F**

ZLIST structure vos programmes avant la sortie écran ou imprimante, sans modifier l'espace mémoire. ZLIST espace chaque mot clés, idente les FOR-NEXT, IF ELSE. Très utile pour la mise au point de vos programmes.
MOD I K7, DISK .... **199 F**

UTILDIR mémorise à votre place le contenu de 80 disquettes. Ne cherchez plus un programme. UTILDIR sait où il se trouve.
MOD I DISK .... **290 F**

GESTION de stocks. GESTION comptes clients pour Modèle I, II, III, de 1000 à 2000 fiches. Standard ou sur mesure. Nous consulter.

* Trademark of Tandy C°

Retourner à : **M. H. POSSARD** - TABANAC - 33550 LANGOIRAN
☎ (56) 67.29.61

Je désire recevoir une documentation détaillée sur ces produits.

NOM : .... Profession : ....
Adresse : .... Code postal : ....
Ville : .... Téléphone : ....

UN TRS-80
ENCORE PLUS PERFORMANT

NOUVEAU ! ACCEL 3

COMPILATEUR BASIC ET DISK BASIC

- Version améliorée et entièrement réécrite d'ACCEL 2, lequel a été plusieurs fois cité, dans la presse américaine, comme étant l'un des meilleurs compilateurs.
- Idéal pour les applications professionnelles.
- Les programmes normaux sont traduits en langage machine sans que des modifications soient nécessaires.
- L'augmentation de la vitesse d'exécution est telle que le programmeur BASIC peut accéder aux domaines réservés du spécialiste en assembleur, par exemple :

— génération de musique
— applications graphiques complexes..... K7 ou disque : 650 F TTC
— **ACCEL** : Version simplifiée d'ACCEL 3.......... K7 : 259 F TTC
— **EDIT** : Editeur BASIC plein écran avec curseur "flottant". Rend l'écriture et la correction des programmes BASIC rapide et agréable. L'utilitaire indispensable du programmeur........... K7 : 230 F TTC
— **TSAVE + DETECT** : permettent de réaliser simplement (sans assembleur, ni hexadécimal) des copies de sauvegarde des bandes en format "SYSTEM" standard.................... K7 : 195 F TTC

* Tous ces programmes pour TRS-80 modèle I et III L2 et Vidéo-Génie à partir de 16K RAM sont accompagnés d'une documentation détaillée en français.
* Demandez notre catalogue pour plus de détails (+ LDOS, ACORN SOFTWARE, etc.).
* Port : ajoutez 20 F (30 F pour l'étranger, 45 F si contre remboursement) à votre commande.

TRS-80 : marque déposée de Tandy

GENERAL SOFTWARE

19, bd Montmartre - 75002 PARIS - Répondeur 531.49.26
VENTE PAR CORRESPONDANCE EXCLUSIVEMENT





Scénario **Y. LEMPEREUR**
Copyright © 1982 by **MICRO 2000**

THE BLACK HOLE :

Les Dorfians viennent d'envahir l'Empire Galactique. La flotte de l'Empire riposte et détruit tous les envahisseurs, à l'exception de leur chef. Celui-ci s'est réfugié au fond d'un "Trou Noir". Votre mission: le détruire!



APPLE PANIC :

D'horribles pommes vous poursuivent. Vous devez creuser des trous dans le sol de briques de manière à les y faire tomber, puis les détruire en les frappant sur la tête.
Si vous n'y arrivez pas à temps, la pomme se hisse hors du trou et... vous dévore!



• Les programmes sont écrits en langage "machine" • Le score le plus élevé est automatiquement mis en mémoire dans la version "disque" de Apple Panic • L'emballage de tous les jeux est en couleurs.

Prix par jeu:	Tape **1.250 FB — 135 FF**	frais de port
	Disk **1.450 FB — 155 FF**	non-compris

• **Nous acceptons les cartes VISA et MASTERCHARGE**
Conditions spéciales pour revendeurs.

Bientôt: "Alien eggs", "Bable terror"...

MICRO 2000

MICRO 2000 sprl
44-46, Avenue Tervueren
1040-Brussels (Belgium-Europe)
Tel. 02/733.65.40

TRACE

le magazine des utilisateurs de TRS-80*

TRS-80 modèle 1, TRS-80 modèle 2, TRS-80 modèle 3, TRS-80 Couleur, TRS-80 de poche, Sharp* PC-1211, Vidéo-Génie*, LNW* : si vous utilisez ou si vous comptez acheter un de ces ordinateurs, sachez que la revue TRACE a été créée pour vous. TRACE vous fournit quatre fois par an des programmes, des astuces, de nouvelles idées d'applications. TRACE est indépendant de tout constructeur, et vous tient au courant de toutes les nouveautés dans les domaines des logiciels, des matériels et des périphériques, quelle qu'en soit la source. TRACE teste pour vous, en toute objectivité et indépendance, les produits matériels ou logiciels qui vous intéressent. TRACE n'est pas en vente chez les marchands de journaux. Pour vous abonner ou pour recevoir un numéro, il vous suffit de nous retourner le bon de commande ci-dessous.



*TRS 80, Sharp, Vidéo-Génie et LNW sont des marques déposées.

TRACE, le complément indispensable de votre TRS-80

ORDI-5

LE MAGAZINE DES UTILISATEURS DE SINCLAIR



Si vous utilisez un ordinateur SINCLAIR (ZX 81, ZX 80 ou Spectrum) ou si vous comptez en acheter un, sachez que la revue **ORDI-5** a été créée pour vous. Indépendant de tout constructeur ou importateur, **ORDI-5** vous fournit quatre fois par an des programmes, des conseils, des astuces, de nouvelles idées d'utilisation. **ORDI-5** teste pour vous en toute objectivité et indépendance les produits matériels et logiciels adaptables sur votre SINCLAIR. **ORDI-5** vous tient au courant de toutes les nouveautés susceptibles de vous intéresser.

ORDI-5 n'est pas en vente chez les marchands de journaux. Pour le recevoir, il vous suffit de nous retourner le bon de commande ci-dessous.
Vous pouvez également vous abonner en profitant de notre **tarif de lancement**.

ORDI-5, le complément indispensable de votre ZX

*marques déposées

TR 3

DES LIVRES POUR VOTRE TRS

LANGAGES

Le Basic et ses fichiers
Tome 1 - méthodes pratiques
par Jacques Boisgontier

P.S.I. Cet ouvrage s'intéresse à la programmation des applications utilisant des fichiers sur disquettes ou sur disques. La version de Basic retenue est le 5. de Microsoft fonctionnant sous CP/M. Les utilisateurs de TRS-80 et de P.S.I. à microprocesseurs Z 80 et TRS 80 sont donc directement intéressés.

Série rouge
144 pages - 75,00 FF / 570,00 FB

Le Basic et ses fichiers
Tome 2 - programmes
par Jacques Boisgontier

P.S.I. Ce second tome est essentiellement consacré à des programmes, utilitaires comme le générateur de saisie d'écran ou le tri rapide, de gestion comme la facturation ou la paie.

Série rouge
160 pages - 75,00 FF / 570,00 FB

Comment Programmer
par Jean-Claude Barbance

P.S.I. Pour ceux qui ont déjà écrit plusieurs programmes et qui veulent s'attaquer à des réalisations plus ambitieuses. Une méthode, illustrée par trois exemples écrits en Basic :
1) Sous-programme de traduction d'un nombre en mots (101 = cent un); 2) Jeu du 421;
3) Comptabilité familiale.

Série rouge
164 pages - 75,00 FF / 570,00 FB

MATÉRIELS

La pratique du TRS-80
Volume 1
par Pierre Giraud et Alain Pinaud

P.S.I. Cet ouvrage s'adresse aux amateurs de TRS-80 Modèles I et III déjà initiés. Il permettra aux utilisateurs de TRS-80 de tirer le meilleur parti de leur ordinateur et, en particulier, du langage BASIC niveau II. Il est complété par des exemples de programmes et des annexes pratiques.

Série bleue
128 pages - 65,00 FF / 500,00 FB

La pratique du TRS-80
Volume 2
par Pierre Giraud et Alain Pinaud

P.S.I. Ce volume s'adresse à ceux qui s'intéressent à la programmation en langage assembleur du Z-80. L'auteur aborde l'aspect mise au point de programmes écrits en langage assembleur et cite quelques outils. Il présente enfin des exemples de sous-programmes et décrit les entrées-sorties du TRS-80.

Série rouge
212 pages - 85,00 FF / 650,00 FB

La pratique du TRS-80
Volume 3
par Pierre Giraud et Alain Pinaud

P.S.I. Après le logiciel, le matériel, ce volume vous guide dans l'exploration systématique (55 schémas) du TRS-80 Modèle I. Pour mordus sans complexe en électronique dont une des armes familières est le fer à souder. Ce livre est, par ailleurs, une documentation originale et complète sur la structure d'un ordinateur.

Série noire
128 pages - 75,00 FF / 570,00 FB

La découverte du PC-1211
par Jean-Pierre Richard

P.S.I. Au fil des chapitres, cet ouvrage fera découvrir au non initié instructions et commandes, variables et mémoires, fonctions périphériques. Enrichi d'exercices d'applications et d'un index, ce manuel fournit à l'utilisateur tous les éléments de base nécessaires à la programmation en langage Basic du PC-1211 (ou TRS-pocket).

Série verte
152 pages - 75,00 FF / 570,00 FB

GUIDES

APL sur TRS 80
par Claude Nowakowski

P.S.I. Consacré aux deux interpréteurs actuellement disponibles pour TRS-80, APL 80 cassette et APL 80 disquette, ce guide pratique démontre que ce langage très évolué mérite l'attention des utilisateurs d'ordinateur individuel. Ils y découvriront en effet que ce langage interprété est actuellement plus efficace que les autres (temps d'exécution et encombrement mémoire).

Série bleue.
96 pages - 65,00 FF / 500,00 FB

TRADUCTIONS

Les graphiques sur TRS-80
par Don Inman
traduit par Alain Pinaud

P.S.I. Dans ce livre, l'auteur montre comment, avec une connaissance minimale du Basic et du TRS-80, visualiser des graphiques sur écran. Il débute par les notions élémentaires de tracés de ligne puis aborde les dessins géométriques pour déboucher sur l'animation de dessin et autres applications plus complexes.

Série verte
144 pages - 75,00 FF / 570,00 FB

Le BASIC et ses Fichiers — Tome 1 méthodes pratiques — Jacques Boisgontier — P.S.I.

Le BASIC et ses Fichiers — Tome 2 programmes — Jacques Boisgontier — P.S.I.

Les graphiques sur TRS-80 — Don Inman — traduit de l'américain par Alain Pinaud — P.S.I.

La pratique du TRS-80 — Volume III - Fonctionnement du matériel Schématique - composants — Pierre Giraud, Alain Pinaud — P.S.I.

Comment Programmer — Jean-Claude Barbance — P.S.I.

La pratique du TRS-80 — Volume II - Compléments Basic II Programmation du Z80 — Pierre Giraud, Alain Pinaud — P.S.I.

La pratique du TRS-80 — Volume I - Architecture et Basic II — Pierre Giraud, Alain Pinaud — P.S.I.

GUIDE PRATIQUE — L'APL SUR TRS-80 — CLAUDE NOWAKOWSKI — ...NS DU P.S.I.

NOUVEAU

la découverte du PC-1211 — jean-pierre richard — P.S.I.

AGRAPH

P.S.I. DIFFUSION

P.S.I. DIFFUSION
41-51, rue Jacquard
BP 86 - 77400 Lagny-s/Marne
FRANCE
Téléphone (6) 007.59.31

P.S.I. BENELUX
5, avenue de la Ferme Rose
1180 Bruxelles
BELGIQUE
Téléphone (2) 345.08.50

BON DE COMMANDE

Envoyer ce bon accompagné de votre règlement à P.S.I. DIFFUSION ou, pour la Belgique et le Luxembourg, à P.S.I. BENELUX

T.TR

DESIGNATION	NOMBRE	PRIX
	TOTAL	

(par avion : ajouter 8 FF (75 FB) par livre)

NOM ____________ PRENOM ____________

rue ____________ N° ____________

Code post. ⊔⊔⊔⊔⊔ Ville ____________

Date et signature :

courrier des lecteurs

Un fil à la patte...

- Le programme "super driver pour imprimante parallèle", paru dans le numéro 1 de TRACE, fonctionne correctement sauf lorsque je désire modifier dynamiquement les paramètres par la commande *PARAL, où le message "file not in directory" est affiché. (Philippe Biessy).

* OK Phil, vous avez trouvé l'erreur ! Le message correct aurait dû être "Phil not in directory" ! (Non, il ne s'agit pas d'un gag !) Vous avez maintenant gagné le droit de choisir entre deux solutions...: soit appeler votre programme : PHIL, soit remplacer, à la ligne 940, le texte "PHIL" par le nom que vous avez donné à votre programme : DEFM 'PARAL...' (ATTENTION : les points représentent trois espaces pour arriver à un total de 8 caractères. Mystérieusement d'ailleurs, l'apostrophe finale a disparu en page 45 de notre numéro 1). Mais pourquoi "PHIL" direz-vous ? Tout simplement parce que ce driver pilotait à l'origine une imprimante Philips...

Editeur/assembleur ou macro-assembleur ?

- Votre article "initiation à l'assembleur" (TRACE n°1) a mis le feu aux poudres... et je me suis empressé d'acheter "l'editor/assembler" version disque, dans un magasin Tandy. Ma stupéfaction était grande en constatant que ma version n'était pas la même que celle présentée : 2 disquettes comprenant EDIT-80, MACRO-80, etc... (R. Sdettel).

* Le programme utilisé dans notre article se nomme EDTASM et est en vente dans les magasins Tandy pour la version cassette uniquement. Ce programme, repris par la firme Apparat, a été adapté à la version disquette, et figure comme tel sur les systèmes NEWDOS et NEWDOS80, mais n'est pas vendu séparément. D'autre part, Microsoft commercialise une version "évoluée" de ce programme, nommé EDTASM +, avec de nombreuses améliorations (macros notamment), et qui supporte les versions cassette et disquette (voir TRACE n°2 pour plus de détails). A notre connaissance, ces deux produits ne se trouvent pas chez Tandy qui, par contre, commercialise le macro-assembleur M80 très puissant, mais d'un emploi plus complexe pour les débutants. Nous en reparlerons un jour !

MOD pas à la mode du Level 2...

- Comment peut-on remplacer l'instruction MOD sur un TRS ? (J.R. Sdettel).

* Sur certains BASIC, on rencontre parfois l'opérateur MOD (module), qui correspond au reste de la division entière de deux valeurs (17 MOD 5, par exemple, rend la valeur 2). Sur les BASIC qui ne le possèdent pas, X MOD Y peut être remplacé par X-Y* FIX(X/Y).

... et le modèle 3 ?

- Pourriez-vous proposer des modifications ou "montages" en ce qui concerne le modèle 3 ? Je suis vivement intéressé par un article qui traiterait de la façon de monter soi-même une unité de disquette. Connaissez-vous une firme qui vend un tel kit, ou un lecteur ayant une telle réalisation à son actif? (C. Cassimans).

* Le modèle 3 est assez récent, et nous ne disposons pas actuellement, de montages spécifiques à ce matériel. Mais que les lecteurs n'hésitent surtout pas à nous soumettre les leurs ! Concernant l'installation de la première unité de disquette, il faut savoir que le "kit" est constitué, en plus de l'unité proprement dite, d'une alimentation (pour deux disquettes), d'une carte contrôleur, d'une mécanique permettant de fixer l'ensemble, du système d'exploitation TRSDOS et de sa documentation, et enfin de trois cables plats. Tandy ne vend pas séparément ces différents sous-ensembles. Aux Etats-Unis, cela est possible auprès de certaines firmes (Percom et MTI notamment ; cette dernière semble avoir un représentant à Paris, qui répond au numéro de téléphone : 306.46.06). Toute société capable de fournir ces composants séparément est invitée à nous contacter. Quant aux lecteurs ayant eu quelque expérience à ce sujet, qu'ils nous en fassent part (un article serait le bienvenu).

- Quelqu'un a-t-il réalisé sur modèle 3, les extensions BASIC du livre "la pratique du TRS-80" volume 2 ? (G. Briand, Club Ademir-Orléans).

* Il serait bien rare que cela ne soit pas déjà fait... Nous attendons les réponses (le cas échéant, il n'est pas trop tard pour se mettre au travail) !

Stringy floppy ?

- Votre revue contient une publicité qui propose un système "plus rapide et plus fiable que les cassettes", et qui se nomme ESF (Exatron Stringy Floppy). Pourriez-vous parler et tester ce produit?

... En effet, la publicité, comme toute publicité, enjolive son produit tout en masquant les côtés néfastes du matériel... ne connaissant pas les disques, je ne suis pas à même de juger. (B. Jouanjus).

* Ce matériel nous a été promis pour un essai, dès qu'il serait disponible. Nous devons malheureusement en conclure qu'il ne l'est toujours pas à l'heure actuelle.

Favoritisme ?

- Vous faites, selon moi, la part trop belle aux possesseurs de DOS. Je n'ai rien contre le DOS naturellement, rien non plus contre les Mercedes, BMW et autres Bentley. Et vous ? Nos épouses convertissent très vite le prix du DOS (et de l'interface), en machines à laver la vaisselle, ou autres appareils électro-ménagers... Donnez-leur la parole, vous verrez bien...(C.Darbaud).

* Ça, c'est une bonne idée : écrivez-nous, mesdames, surtout si vous êtes des "fanas" du TRS ! Voyons, ne vous aurait-on pas appris à partager : une machine à laver, un TRS, un magnétoscope, une disquette... Et puis, avec l'ordinateur au foyer, plus de mari au bistrot, et plus de gosses au flipper...! La télé ? Mais chacun sait que c'est très mauvais pour les enfants ! L'ordinateur, lui, il fait son entrée dans les écoles. Alors, vous voyez ! Pensez à l'avenir de vos enfants. De plus, il est impossible d'apprendre correctement l'anglais ou les mathématiques chez soi, sans disquette...(Nous avons fait notre possible, monsieur Darbaud)!

Tactique ou stratégie ? (lettre ouverte)
------------ -------------------------

- Comment faire pour acheter un logiciel de jeu ? Naïvement, il m'avait semblé que les boutiques d'informatique étaient là pour offrir ce type de service. Aussi, ayant porté mon choix sur un jeu de stratégie, je téléphonai à la boutique parisienne X pour m'assurer de la disponibilité de ce programme et de la possibilité d'avoir une démonstration. Avant de dépenser 450 frs, il était préférable de vérifier l'intérêt du jeu ainsi que la manière dont était représenté graphiquement le terrain.
Rendez-vous étant pris, je me présentai un samedi matin, assisté de mes deux fils, spécialistes en la matière. Après trois quarts d'heure d'attente, quelle ne fut pas notre stupeur devant la réaction du technicien enfin mis à notre disposition. Il refusait tout simplement de faire la démonstration, prétextant le temps nécessaire au chargement du programme (5 minutes!), le risque d'erreur de lecture de la cassette (tiens, tiens!) et la foule du samedi (trois clients dans le magasin). Inutile de préciser que nous sommes repartis sans avoir fait le moindre achat tout en ayant fait remarquer au responsable de la boutique l'incongruité de faire déplacer des clients pour finalement les decevoir à ce point.

Finalement nous avons attendu la parution d'un banc d'essai dans une revue pour acheter dans une autre boutique le jeu de stratégie Tigers in the snow (la bataille des Ardennes). Hélas, une erreur de lecture de la cassette rendait le programme inutilisable. Après retour de celle-ci au vendeur (très aimable), il m'a fallu quinze jours pour disposer d'une nouvelle copie. Mais cette fois-ci, le fichier de données, écrit sur l'autre face, n'était plus lisible !

Ayant acheté par correspondance aux Etats-Unis, plusieurs programmes sur cassette sans le moindre ennui et à des prix autrement compétitifs, l'intérêt de passer par une boutique, dans ces conditions de services, ne m'apparait pas évident. (L.P.).

* Vous semblez effectivement avoir accumulé les déconvenues. En ce qui concerne l'accueil dédaigneux reçu dans la première boutique, tout comme celui aimable que vous avez trouvé dans la seconde, il est difficile d'en tirer des conclusions trop définitives : il suffit que vous tombiez sur un vendeur débutant et peu compétent pour rencontrer souvent un accueil sans la moindre considération, alors que dans la même boutique l'accueil d'un autre vendeur (ou du même un autre jour) sera très agréable. Regardez bien autour de vous : vous rencontrez le même genre d'accueil "variable" dans toutes les boutiques, quoi qu'elles vendent - pain, viande, hi-fi, livres, disques, vêtements, etc. - Faites pour vos programmes comme pour vos autres achats : retournez là où vous avez rencontré le meilleur accueil et le meilleur service, et essayez d'avoir le même vendeur. Les gens peu aimables ou peu serviables s'éliminent ainsi d'eux-mêmes, alors que les autres reçoivent un juste retour à leurs efforts.
En ce qui concerne d'autres sources d'approvisionnement, vous avez raison d'insister sur le fait que si les boutiques ne donnent pas un bon service, il vaut mieux acheter par correspondance en France ou aux Etats-Unis. Et il faut également ajouter que si les utilisateurs ne peuvent essayer les programmes, ceci risque de leur donner bonne conscience s'ils en "empruntent" une copie plus ou moins pirate à un ami.

MAGAZINE

CP/M-86 POUR MODELE 2

Alors que Tandy annonce le modèle 16 aux Etats-Unis, ainsi qu'un kit permettant de transformer le modèle 2 en modèle 16 (microprocesseur 16 bits 68000 de Motorola), la firme californienne Veritas Technology Inc propose une autre solution, très séduisante : une carte bi-processeur équipée d'un Z80, d'un 8086 (16 bits Intel) et de 128K octets de mémoire (extensible à 256K), occupant un seul emplacement dans le chassis du TRS (prix : de l'ordre de 1500 dollars environ). Avec cette option, le modèle 2 peut ainsi recevoir les systèmes d'exploitation CP/M-86 (Digital Research) et MS-DOS (Microsoft) équipant les ordinateurs individuels IBM et SIRIUS notamment, tout en restant compatible, bien entendu, avec CP/M-80 et TRSDOS. Voilà une direction qui nous semble moins tortueuse que celle adoptée par Tandy, du moins, tant que nous ne connaissons pas mieux le modèle 16 ...

NOUVEAUX LOGICIELS

La Société General Software annonce deux produits très intéressants sur modèles 1 et 3 : ACCEL3 et Small-LDOS. Le premier est une version améliorée et entièrement réécrite du compilateur ACCEL2 (probablement déjà connu de nos lecteurs). Par rapport à celui-ci, la taille des programmes produits est réduite, ainsi que la durée de compilation, alors que le nombre des fonctions traduites en code machine est augmenté. Les instructions inconnues d'ACCEL3 continuent d'être exécutées par l'interprêteur. Enfin, il est possible de sauver directement sur disquette ou cassette, le code généré. Prix TTC avec manuel en français : 650 F.

Small-LDOS est une version simplifiée du célèbre système d'exploitation de disquettes : LDOS, dont il conserve les meilleures commandes, ainsi que l'intégralité du LBASIC. Ce système de taille réduite, est à conseiller pour les utilisations courantes ne nécessitant pas le niveau de sophistication du LDOS. Prix TTC (manuel anglais) : 830 F.

VUS A MICRO-EXPO

A l'occasion de Micro-expo, Tandy vient d'annoncer un disque dur pour le modèle 2, d'une capacité de 8,4 millions d'octets. Compatible avec la plupart des logiciels existants, il est possible de connecter jusqu'à quatre unités semblables (33 Moctets) sur cet ordinateur.
Cet organe périphérique, de technologie Winchester, est en fait composé de deux disques 8 pouces situés dans un milieu hermétique à la poussière, offrant donc quatre faces magnétiques de 256 pistes chacune. La vitesse de transfert est de 4,34 millions de bits par seconde, pour une vitesse de rotation de 3125 tr/mn. Le temps d'accès piste à piste est de 19 millisecondes.

Un grand revenant sur le stand Tandy : le TRS couleur (Coco pour les intimes) ... qui, cette fois, est équipé d'une prise Péritel, ce qui lui permet de s'adapter sans le moindre problème aux téléviseurs couleur munis de cette possibilité de connexion (les postes récents en sont systématiquement pourvus). Voilà qui va sérieusement descendre le prix de la configuration de base, puisque les utilisateurs ne seront plus obligés d'acheter également le téléviseur au standard PAL ou NTSC, compagnon inséparable - jusqu'à maintenant - du TRS couleur français.

Contre toute attente, le modèle 16 (version 16 bits du modèle 2) n'était pas visible chez son constructeur, mais à quelques mètres de là, sur le stand "Espace 01"... la "bête" ayant été directement importée des Etats-Unis. Il est vrai que le système, dont on pouvait toutefois admirer l'esthétique, ne brillait pas par la démonstrativité de son logiciel, dont il semblait presque totalement démuni. Enfin, nous sommes heureux d'avoir constaté qu'il existait. Mais qu'en est-il du prix et des délais de livraison ?

Déplaçons-nous encore de quelques mètres, où la Société MDD (Micro Défense Diffusion) présentait un modèle 3 reconditionné aux Etats-Unis par les bons soins de la firme MTI. A part sa couleur blanche, rien ne le distingue du modèle standard. Pourtant, une fois sous tension, les résultats sont très significatifs : horloge 4 Mhz (au lieu de 2), deux modes de fonctionnement selon le système d'exploitation DOSPLUS 4.0 (compatible TRSDOS) ou CP/M 2.2. Sous le contrôle de ce dernier, il est possible au moyen d'un ordre simple, de passer l'affichage en 24 lignes de 80 caractères ! En ce qui concerne les périphériques magnétiques, le choix est grand : 40 pistes, 80 pistes, double face, double densité (jusqu'à 1,5 Moctets), et même disque dur 5 pouces externe ou se logeant en place d'une minidisquette (5 ou 7,5 Mo) ... Selon la configuration, la gamme comprend quatre modèles .
Chacun d'eux comprend au moins l'horloge rapide ainsi qu'un système de ventilation.

GES présentait également le Vidéo Génie 3, qui, comme le modèle précédent, a l'avantage d'accepter, en plus du système d'exploitation standard NEWDOS-80, le CP/M 2.2 avec un format de 24 lignes de 80 caractères. Comme le modèle 3, le système se présente sous la forme d'un ensemble intégré comportant un écran, un clavier et deux unités de minidisquettes.

Dans le cadre de Microexpo se tenait le jeudi 17 juin une session consacrée aux TRS-80.
Indiscutablement cette demi-journée a été un succès. Devant une salle comble, huit orateurs ont fait le point de leurs expériences et de leurs connaissances des TRS-80 actuels et futurs.

D'une manière générale, les orateurs ainsi que les participants qui ont eu à se prononcer sur la qualité de leur matériel ont littéralement encensé leur TRS-80. Mais il est vrai que dans ce genre de situation, rares sont les orateurs qui racontent leurs déboires !

Présidée par M. Montanari, le Président de l'Association des Utilisateurs de TRS-80 (AUT), la session a débuté par un exposé de M. Le Flour, professeur de

mathématiques, qui a indiqué que, selon lui, les TRS sont grâce à leur prix les ordinateurs individuels les mieux adaptés à l'enseignement. Il a cité quelques établissements pédagogiques où ces matériels sont utilisés : le lycée Voltaire, la faculté de sciences de Jussieu, la faculté de Vincennes, l'INEP à Marly le Roi, la Chambre de Commerce du Loiret, etc. Parmi les produits pédagogiques intéressants, il a présenté Eurydice qui est un langage - auteur sur modèle 3 - dont la caractéristique est de permettre à des enseignants non informaticiens de mettre au point, pour leurs élèves, des cours assistés par ordinateur - et Studial qui est un système d'apprentissage du clavier réalisé par la société Pigier.

Ce fut ensuite au tour de M. Civet, ingénieur-conseil en informatique, de présenter une très intéressante utilisation d'un modèle 3 dans un laboratoire d'analyse médical. Parmi les applications réalisées, l'automatisation d'une petite paie (21 bulletins), de la comptabilité (500 lignes par mois), et de la cytologie (800 frottis par mois). Ce système a évité au laboratoire de recruter une personne de plus, il a permis d'augmenter la production, il a donné de bons outils de gestion aux responsables. Tout cela pour environ 27 000 francs ttc de matériel et moins de 3 mois-homme d'analyse et programmation, pratiquement entièrement en Basic.

P. Giraud, responsable des produits chez Tandy France a exposé alors quelques utilisations possibles des modèles 2 comme terminaux compatibles de la "grosse" informatique (IBM 3270 ou 2780) et dans les réseaux. L'accent a été mis sur Tandy Dial, un nouveau service apparenté au réseau Micro Dial. On peut toutefois s'interroger sur le succès d'un tel service tant que les modems coûtent plus de 3000 FF ttc.

Ensuite M. Mignier et M. Merck ont présenté avec un film les conditions dans lesquelles l'informatique individuelle a été introduite dans une grosse entreprise de 2000 salariés (en France), à savoir Moore Paragon. Au Sicob 1979, M. Merck, Directeur des Relations Humaines, a eu le coup de foudre pour l'informatique individuelle. Il est reparti avec un modèle 1 sous le bras. Il a très rapidement convaincu son PDG que ce nouvel outil devrait permettre de développer dans l'entreprise l'autonomie et la créativité des membres du personnel. Et, aujourd'hui, 25 modèles 1 ou 3 fonctionnent plus de 8 heures par jour. Plus de 300 programmes ont été écrits par leurs propres utilisateurs. Chaque salarié peut suivre deux journées d'initiation au Basic et à VisiCalc. Chaque service peut ensuite acquérir un TRS-80 sur son propre budget, et non pas sur celui de la Direction Informatique. Les ordinateurs individuels sont utilisés à des fins multiples : statistiques, budgets, états de stocks, comptes d'exploitation, calculs de prix, etc.

Une anecdote sur la fiabilité du modèle 3 : un utilisateur avait de longues listes à imprimer, et il décida d'aller déjeuner en laissant son système travailler tout seul, les fenêtres grandes ouvertes. Quelques minutes plus tard, le jardinier installa un système d'arrosage automatique, puis alla vaquer ailleurs à d'autres occupations. A son retour, l'utilisateur eut la mauvaise surprise de trouver son modèle 3 et ses listes complètement mouillés. Mais bien heureusement tout continuait à fonctionner normalement !

Après l'expérience Moore Paragon, un autre ingénieur-conseil, M. Faure, parla de quelques applications mises en place dans des entreprises de toutes tailles La plus spectaculaire est peut-être celle réalisée dans une entreprise. Disposant d'un gros IBM 360/40, elle envisageait de commander un non moins gros IBM 370/148 pour le remplacer. Finalement, ce furent 12 modèles 2 qui ont été choisis à la place, permettant ainsi une plus grande souplesse et une sérieuse économie.
Selon M.Faure, dès qu'un ordinateur individuel est utilisé plus de 2 heures par jour, il faut envisager d'en commander un autre.

M. Lamoitier a présenté quelques systèmes d'exploitation disponibles notamment sur modèle 2 et en particulier CP/M et le p-system (UCSD) qui sont adaptés à des configurations monopostes, et Oasis, davantage orienté multiposte.

Un des participants ayant poussé M. Montanari dans ses derniers retranchements, celui-ci a avoué que le système d'exploitation qu'il préconise sur modèle 1 et modèle 3 est le New DOS 80 version 2.

Après cela, un film a été projeté pour présenter la société Tandy France (était-ce bien utile, les participants étant des connaisseurs, voire des convaincus?) et le Directeur Commercial, M. Beaufort, a annoncé les nouveaux produits disponibles dès à présent ou très prochainement sur le marché français. Il est ainsi possible de connecter un TRS couleur sur un téléviseur Secam (enfin) doté d'une prise péritel ; Scripsit fonctionne sur modèle 3 avec un clavier AZERTY (re-enfin!) ; le modèle 2 peut être connecté simultanément à 4 disquettes de 500 000 caractères, 1 disque dur de 8,4 MO et 3 disques durs de 8,9 MO.

Avant la fin de l'année, le modèle 16 sera commercialisé sur le marché français (sans doute avec guère plus de logiciel qu'aux Etats-Unis), ainsi que les terminaux DT1 (qui sont des modèles 3 sans mémoire) et un réseau local semblable à Arcnet.

Michel Dunant

DERNIERE MINUTE :

Tandy annonce la couleur !

Début juin, nous avons été surpris de voir dans les boutiques américaines que le TRS-couleur 4K Niveau 1 (référence 26-3001) était soldé à 299 $ (2000 FF environ) au lieu du prix catalogue de 399 $ (2700 FF), sans moniteur vidéo couleur bien sûr. D'après les différents responsables de boutiques que nous avons alors pu interroger, c'était parce que la vente de ce modèle allait être arrêtée, du moins aux Etats-Unis, et qu'il fallait donc "vider les stocks".

Effectivement, Tandy a depuis annoncé mi-juin aux Etats-Unis sa nouvelle "version de base", ré-

MAGAZINE

férencée 26-3004 ; elle dispose en standard de 16K de mémoire vive et d'un "BASIC couleur", et coûte 399 $ (2700 FF environ). Il ne s'agit cependant vraisemblablement pas là du même BASIC que le "BASIC couleur étendu" de référence 26-3018, vendu aux Etats-Unis 99 $ (670 FF), et qui nécessite lui aussi 16K de mémoire ; ce BASIC est (était?) disponible en standard sur les versions 16K de référence 26-3002 (599 $, soit environ 4100 FF), et 32K de référence 26-3003 (699 $, soit environ 4800 FF).

En attendant l'annonce en France du 26-3004, le "vieux couleur" 26-3001, maintenant disponible avec interface couleur, est vendu 4295 FF ttc, l'extension à 16K coûtant 675 FF ttc (soit le même prix que cette extension aux Etats-Unis!!!) et les mémoires mortes du BASIC couleur étendu coûtent 1125 FF ttc (alors qu'aux Etats-Unis elles coûtent le même prix que l'extension à 16K de mémoire vive).

En même temps que le "nouveau couleur", Tandy a également annoncé, toujours aux Etats-Unis, une imprimante graphique 4 couleurs et un terminal imprimant portable.

L'imprimante graphique, référence 26-92, sera livrée fin 82. Elle permet d'imprimer 4 couleurs sur 80 ou 40 caractères/ligne à la vitesse de 12 caractères/seconde. Les interfaces série et parallèle sont intégrées. Le prix ? seulement 250 $ (1700 FF ttc), ce qui devrait"faire un malheur".

Le terminal portable, référencé PT-210, comporte un clavier à 67 touches, une imprimante thermique 80 colonnes 50 caractères/seconde et un modem acoustique intégré. Prix : 995 $ (6800 FF environ), avec en option une interface série (70 $, soit 480 FF environ). Avec cette dernière option, on peut utiliser le PT-210 comme terminal ou imprimante d'un ordinateur quelconque : pas mal ! Notez toutefois que dans l'immédiat, seule cette dernière utilisation serait possible en France, le modem acoustique étant aux noms américains, et non européens (...non compatibles...). De plus, Tandy ne vendra pas ce terminal en France sans avoir fait homologuer le modem par les PTT, même si les modems acoustiques ne sont pas vraiment soumis à l'homologation obligatoire.

TANDY vient de créer un système d'assistance téléphonique, que les utilisateurs de TRS peuvent appeler "pour obtenir tout renseignement nécéssaire à une meilleur utilisation de leur TRS". Nous n'avons pas encore pu tester l'efficacité et l'amabilité des personnes qui assurent cette assistance téléphonique, mais que nos lecteurs nous fassent part de leur expérience, qu'elle soit bonne ou mauvaise.

Apparemment les utilisateurs appellent ces numéros à leurs frais. Sans doute que TANDY envisage très vite de s'abonner au "Service Libre Appel" du téléphone : la satisfaction des utilisateurs n'a pas de prix, n'est-ce pas ? En attendant, on peut sans doute essayer d'appeler en PCV...

En France : (1) 238 84 29
En Belgique : (2) 647 10 91

Aux Etats-Unis, les versions recarrossées des différents TRS commencent à apparaitre dans les boutiques. Dans certains cas, l'initiative n'appartient pas à TANDY, mais à ses clients tels que MTI qui achète des Modèle 3 en quantité et leur adjoint différents développements matériels. Mais la grande nouvauté, c'est que maintenant TANDY le fait, par exemple pour le couleur que l'on commence à voir, dans d'autres boutiques que les Radio-Shack, avec une carrosserie blanche et un Cogo légèrement différent.
L'explication? TANDY a décidé de remettre en cause l'exclusivité de son réseau de boutiques, et de le compléter par des distributeurs indépendants. Ceci n'est valable pour l'instant qu'aux Etats-Unis (On parle de 60 distributeurs pour commencer), mais se généralisera sans doute aux autres pays : un bon moyen de complèter un réseau que TANDY semble actuellement s'essouffler un peu à complèter. Et comme les concurrents du couleur semblent nombreux et bien armés, pourquoi refuser ce bon moyen?
Ce changement radical dans la politique de diffusion de TANDY sera sans doute assez contreversé chez les revendeurs existants, et notamment les franchisés. (Mais n'est-il pas question que TANDY emploie de moins en moins de franchisés?) Et quant à toutes les boutiques françaises indépendantes qui auraient aimé commercialiser les TRS dans des conditions normales (ce qu'elles n'ont jamais pu faire), leur réaction sera-t-elle favorable??

Adresses des sociétés mentionnées

Véritas Technology Inc, 2375 Zanker Rd, Suite 245, San José, California 95131.

General Software, 19 boulevard Montmartre, 75002 Paris.

Tandy, 211-213 boulevard Mac Donald, 75019 Paris.

Espace-01, 7 rue de l'église, 92200 Neuilly

MDD, 82 galerie des Damiers, Défense 1, 92400 Courbevoie

GES, 68, 76 avenue Ledru Rollin, 75012 Paris.

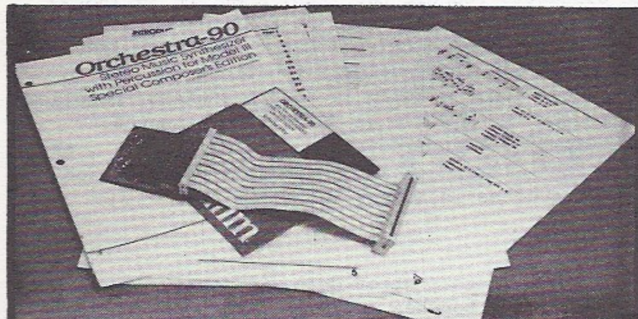
attention
TRACE n° 1
est épuisé

La stéréo sur Modèle 3 : Orchestra 90

Nom : Orchestra-90
Configuration : TRS-80 modèle 3, 16K cassette ou disquette
Constructeur : Software Affair
Distributeur : Graphie France
Prix : 1500 F TTC

Le numéro 1 de TRACE nous avait appris ce qu'était Orchestra-80 : un synthétiseur de musique pour TRS-80 modèle 1, s'adaptant sur la sortie bus de l'ordinateur. Aujourd'hui, nous parlerons d'un appareil similaire, conçu par la même firme (Software Affair Ltd) pour s'adapter sur le modèle 3, mais offrant deux possibilités très intéressantes : le son stéréophonique et des effets de percussion. Il est à noter qu'un modèle semblable existe également sur modèle 1, et se nomme Orchestra-85.

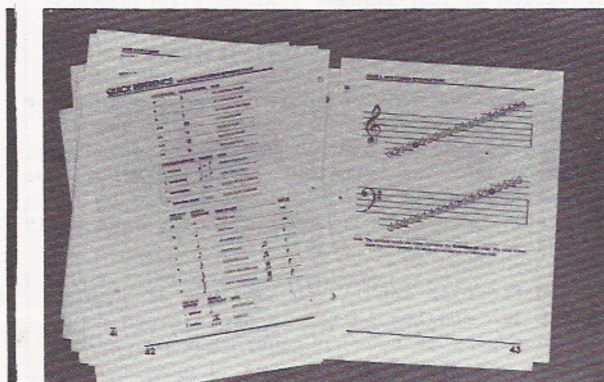
L'ensemble Orchestra-90 est formé d'une petite carte supportant les composants du synthétiseur, terminée par un câble venant s'enficher sur la sortie bus du TRS-80 modèle 3, d'un manuel en anglais (45 pages), et d'un programme fourni sur minidisquette. Une version cassette existe également. Dans les deux cas, une configuration de 16K de mémoire vive est suffisante.



Orchestra-90 permet la génération de cinq "voix" simultanées (autrement dit, cinq sons), mais cette configuration n'est conseillée que si votre modèle 3 est équipée de l'option "horloge rapide 4Mhz". Autrement, un maximum de quatre voix seulement est souhaitable, au-delà duquel la courbe de réponse des fréquences aigues serait trop limitée.

Comme dans le cas d'Orchestra-80, ce synthétiseur vocal ne nécessite pas d'alimentation extérieure, ses circuits électroniques de faible consommation sont alimentés par les signaux fournis sur le connecteur bus. La carte imprimée, d'une taille de 66 x 77 mm, comprend 5 circuits intégrés, 4 boîtiers de résistances et trois condensateurs. Deux prises femelles type RCA sont utilisées pour raccorder le synthétiseur aux entrées stéréophoniques "haut niveau" (100 à 200 mV) d'un amplificateur (entrées tunner, AUX ou magnétophone, par exemple). Ce câble n'est pas fourni.

Orchestra-90 est relié au modèle 3 par l'intermédiaire d'un câble plat de 16 centimètres, au bout duquel est serti un connecteur encartable. ATTENTION : tous les raccordements doivent être effectués avant la mise sous tension du TRS et de l'amplificateur, afin d'éviter des différences de potentiels toujours possibles entre les masses des deux appareils.



En ce qui concerne le logiciel, le programme ORCH90/CMD est composé - comme dans le cas d'Orchestra-80 - d'un éditeur (création des partitions musicales dans un langage simple), d'un compilateur (transcription des partitions en modules exécutables par le synthétiseur) et d'un système de gestion de fichiers (lecture et écriture des partitions sur cassettes ou minidisquettes). Un module d'initialisation permet également de définir des options comme le timbre, le nombre de voix, la vitesse d'exécution, etc.

Un utilitaire (ORCHUTIL/CMD) permet la conversion des programmes musicaux entre supports (cassettes/disquettes) et entre modèles (modèle 1/modèle 3). Mais le plus impressionnant, c'est la bibliothèque des exemples musicaux, composée de 45 morceaux, tous fournis sur la disquette. Et il y en a pour tous les goûts : Carmen, le Vol du Bourdon, Swanee, Yesterday, la Guerre des Etoiles, les Aventuriers de l'Arche Perdue, etc... J'ai particulièrement apprécié (mais c'est une question de goût!) "Blues" avec son accompagnement de contrebasse très réaliste. Vraiment, la stéréophonie et les effets de percussion donnent une toute autre dimension à ce synthétiseur. Après Orchestra-90, on a beaucoup de mal à revenir à Orchestra-80! Un seul reproche toutefois : il présente le même défaut que ce dernier car il n'y a pas de possibilité de moduler la puissance sonore (effet "boîte à musique").

Pour les mélomanes qui ne veulent pas se fatiguer à transcrire des airs connus, on peut se procurer une autre disquette nommée "Greatest Hits",commercialisée par Software Affair, et qui contient 22 morceaux classiques !

Jean-Pierre du Tillet

Dites-le avec l'accent belge... ... ou l'accent texan

Après l'euphorie des premiers jours, les possesseurs de TRS-80 sont rapidement génés par l'impossibilité d'afficher des caractères minuscules à l'écran. Si le maniement du fer à souder vous est familier, une transformation simple permet de résoudre ce problème Le cas échéant, un "kit minuscule" est disponible chez Tandy qui se charge de l'installer. Cependant le problème n'est que partiellement résolu : les caractères accentués utilisés en français n'apparaissent toujours pas ...

D'où l'intérêt de la solution proposée par la société belge BCM : un générateur de caractères qui vient s'emboiter en lieu et place du circuit d'origine. Pour la première fois, j'ai osé me lancer dans la modification de mon système ! Après avoir démonté le clavier, il est enfantin de repérer le circuit Z29 (le seul comportant 18 pattes), de le sortir avec l'aide d'un tournevis (attention, ce circuit est soudé sur les modèles anciens), puis d'enficher à la place le connecteur 16 broches relié par un câble plat au nouveau circuit enrobé de résine (pour décourager les imitateurs?). Il ne reste plus qu'à remettre en place le clavier pour que le système soit prêt à fonctionner, le tout sans avoir eu à souder quoi que ce soit !

Cette modification est livrée avec une disquette contenant les modules logiciels nécessaires pour mettre en oeuvre ce nouveau générateur de caractères soit sous TRSDOS, soit sous le programme de traitement de textes ELECTRIC PENCIL (la touche de fonction est alors simulée par la touche CLEAR), soit sous BASIC. Les instructions accompagnant la documentation permettent de générer très simplement et d'une manière automatique les versions modifiées de TRSDOS et de PENCIL, incluant le module de gestion des caractères accentués, alors qu'un module auto-translatable donne la même fonction sous BASIC. (Je n'ai toutefois pas réussi à obtenir le caractère "oe colle" avec le nouveau PENCIL).

Si vous avez la chance de posséder une imprimante disposant des caractères accentués, deux modules (compatibles NEWDOS 2.1) sont fournis pour imprimer soit avec une DAISY Printer II, soit avec une Centronics 737-2.

La documentation comprend deux pages d'explications, complètes et faciles à comprendre, un fichier PENCIL sur disquette pour les informations complémentaires sur le logiciel, ainsi qu'un jeu d'auto-collants pour marquer les touches commandant les caractères accentués, les commandes de PENCIL, et pour transformer le clavier QWERTY en clavier AZERTY.

Pierre Landry

PRIX : 375 FF + 10,70 FF (Par mandat international)

FOURNISSEUR

(1) B C M (Bureau de Calcul Mathématique)
24 Route de la Sapinière
4960 BANNEUX Belgique
No Banque : 240-0622552-10

Pour ceux qui utilisent le traitement de textes SCRIPSIT, le modèle I peut être amélioré considérablement en installant le kit des minuscules accentuées, maintenant vendu par TANDY (795 F) ; ce kit est le même que celui offert par BCM qui en est d'ailleurs le fabricant. (Voir ci-dessus).

Il est constitué de deux parties : matériel et logiciel.

Le matériel fourni par TANDY est un circuit intégré générateur de caractères modifié, enrobé complètement dans la résine verte ; un câble en nappe le relie à un connecteur mâle qui s'enfiche à la place du générateur de caractères minuscules standard. Un petit interrupteur est relié par 2 fils au boîtier vert, et se pose sur le châssis du TRS, à l'arrière: il suffit de percer un petit trou ou d'agrandir légèrement une des ouïes d'aération du boîtier. Il n'y a aucune soudure à effectuer, ni de trace de circuit à gratter.

Un jeu d'étiquettes autocollantes est fourni ; chaque étiquette se place sur les touches du clavier qui sont réaffectées, certaines se placent en face avant ; au total 42 étiquettes sont placées qui transforment votre clavier en un véritable clavier "AZERTY" standard français : tous les accents sont générés, ainsi que le ç et le symbole §, et s'affichent sur l'écran.

Il faut à peu près une heure de travail pour démonter le clavier, installer le kit, remonter le clavier et coller proprement les étiquettes.

La disquette fournie par TANDY comprend le TRSDOS 2.3, BASIC, et une version de SCRIPSIT adaptée au nouveau clavier : SCRIPSIT/LCA (Lower Case Azerty) ; avec SCRIPSIT/LCA, quelques fonctions ont été déplacées : par exemple la touche <CLEAR> remplace la touche <CONTROL> de SCRIPSIT, la touche <BREAK> est remplacée par <SHIFT BREAK> , etc... Cependant, la réadaptation est très rapide grâce aux étiquettes indiquant les fonctions.

Attention, si vous étiez dejà utilisateur de SCRIPSIT, les touches <A> et <Q> sont inversées, ainsi que les touches <Z> et <W> . N'oubliez donc pas de permuter aussi les auto-collants d'origine de SCRIPSIT : <BLOCK> , <WORD> , <WINDOW> et la flèche à gauche.

Le texte avec ses minuscules accentuées est ensuite édité sur l'imprimante TANDY à marguerite DW II avec une vitesse et une qualité tout à fait professionnelles.

Deux problèmes cependant : si vous avez une autre imprimante que la DW II, vous ne pouvez pas utiliser ce kit pour les éditions car les minuscules sont transformées en caractères graphiques ou en caractères de fonction.

Ensuite, dans les versions de SCRIPSIT accentué jusqu'ici livrées par TANDY, une surprise désagréable : il n'est plus possible d'arrêter l'impression en cours de route ; cette fonction était assurée par <CLEAR> sur SCRIPSIT normal,

et cette touche est devenue la fonction 〈CONTROL〉 : la fonction d'arrêt d'impression devrait maintenant être assurée par les touches 〈SHIFT-CLEAR〉 ; eh bien ! Evitez-la car le programme se "plante" et on est obligé de recharger tout : le DOS, SCRIPSIT, et votre texte si, par chance, il était sauvegardé.

Cette "erreur" du programme est connue des magasins TANDY, mais il n'y a pas encore de solution; disent-ils. TANDY-FRANCE et même TANDY-EUROPE-NANINE, eux, disent tout ignorer du problème : quand donc prendra t-on les utilisateurs au sérieux ?

Si vous êtes la malheureuse victime de SCRIPSIT/LCA, essayez le programme en BASIC ci-joint qui semble apporter la solution : il modifie 5 octets de SCRIPSIT/LCA et nous a été fourni par le créateur du kit, le Bureau C. MICHEL.

Encore une recommandation : si vous utilisez d'autres programmes sur votre TRS, ou d'autres S E D , ils fonctionneront normalement avec l'interrupteur sur la position AR (AZERTY RESTREINT), c'est-à-dire en clavier anglo-saxon QWERTY. Or, si les touches sont recouvertes entièrement avec les adhésifs, vous ne savez plus où est le A, où est le M, etc... alors, avant de coller, découpez l'étiquette en 2 pour conserver quand même les gravures d'origine apparentes !

CONCLUSION

L'ensemble TRS-80 MOD-1 avec 2 unités de disquettes, 32K de mémoire et l'imprimante à marguerite DW II forme un poste complet de traitement de textes de qualité professionnelle pour un coût inférieur à 26 000 F HT. On peut regretter qu'il ait fallu attendre si longtemps pour que les claviers soient francisés alors que bien des modèles concurrents, même américains, sont vendus en France avec un clavier AZERTY. A quand le kit pour modèle II, modèle III, Color ?

Pierre BREHON

NDLR : Depuis l'écriture de cet article, un kit similaire est disponible pour le modèle 3, apparemment d'origine BCM lui aussi. Si nos lecteurs pouvaient nous donner leur opinion sur ce produit...

```
100 DATA50,232,55,211,235
110 F$="SCRIPSIT/CMD"
120 C$=STRING$(2,0)
130 CLS:PRINT"SCRIPSIT        MISE A JOUR 1.0":PRINT
140 OPEN"I",1,F$
150 CLOSE1
160 OPEN"R",1,F$
170 GET1,22
180 FIELD1,122ASA$,5ASE$
190 IFRIGHT$(E$,2)<>C$THEN300
200 C$=""
210 FORI=1TO5
220 READJ
230 C$=C$+CHR$(J)
240 NEXTI
250 LSETE$=C$
260 PUT1,22
270 CLOSE
280 PRINT"MODIFICATION TERMINEE"
290 END
300 PRINT"PROGRAMME NON CONFORME"
310 STOP
```

language

APL sur TRS pourquoi pas !

En fouillant dans les rayons d'une boutique informatique, j'ai découvert une merveilleuse cassette : APL 80, le langage APL pour TRS-80 (modèle I,16K). Cela m'a rappelé la présentation vers 1975, du "miniordinateur" canadien MCM70 qui travaillait en APL. Ce système m'avait fortement impressionné : d'une part, je voyais pour la première fois un ordinateur portable et disposant de performances assez proches du "vrai" ordinateur IBM1130 que j'utilisais alors pour des calculs scientifiques ; d'autre part, j'avais ainsi facilement accès aux possibilités du langage APL qui dans mon esprit était jusque là resté plus ou moins comme une curiosité scientifique sans applications vraiment concrètes. En effet, comme tous ceux qui ont quelques notions de programmation en langage symbolique (Fortran, PL/1, Basic..), j'avais jusqu'alors été tellement surpris lors de l'examen de listings APL que je n'avais pas eu envie d'étudier ce mystérieux dialecte.

Aujourd'hui, je pense que le défaut majeur du langage de programmation APL, c'est qu'il est arrivé trop tôt et que ce langage parait encore à ce jour en avance sur les autres.

Pour une dépense relativement modeste (environ 150 F), j'ai essayé la programmation en APL et au bout de quelques heures, j'ai bien compris de quoi il s'agissait : malgré les limites imposées par un modèle I, 16K, et la cassette APL-80, l'expérience n'en est pas moins très intéressante. Evidemment APL 80 n'est pas APL puisqu'on ne dispose pas du clavier avec les caractères spécifiques à ce langage, mais ce n'est pas cette série de symboles bizarres qui fait sa puissance.

Je ne vais bien sûr pas essayer ici de présenter tout ce qu'on peut faire avec APL 80 (ou APL) ; les quelques exemples qui sont présentés ont pour but de montrer le plus grand paradoxe d'APL : ce

langage est à la fois très simple et très compliqué. On construit rapidement des programmes puissants, capables de traiter des problèmes complexes ; toutefois seule la connaissance des caractéristiques de toutes les fonctions et surtout la pratique permettent de réaliser des programmes concis et très performants. En contre-partie, ces programmes ont, avouons-le, une relecture ou une reprise pour des modifications, souvent assez pénibles.

Lorsqu'on aborde un nouveau langage, on essaie de programmer par exemple : écrire "bonjour", calculer 3+4, faire la somme des 10 premiers nombres, calculer la moyenne d'une suite de nombres...

En APL 80, on frappe "BONJOUR" et le mot BONJOUR s'affiche immédiatement (au lieu de PRINT "BONJOUR" en Basic), et on frappe 3+4, le résultat apparait à la ligne suivante (au lieu de PRINT 3+4). Si on veut sauvegarder le résultat, il suffit d'affecter ce nombre à un nom symbolique :

```
SOMME ⟵ 7
```

Le caractère spécial ⟵ permet donc l'affectation et le nom symbolique peut comporter de 1 à plus de 50 caractères (maximum une ligne). Il est également possible d'affecter d'abord le résultat (implicite) de l'expression :

```
SOMME ⟵ 3+4
```

Pour connaître la valeur affectée à SOMME, il suffit alors de frapper ce nom :

```
SOMME

7
```

La machine APL, en mode exécution, ne connait que ces 2 types d'instructions :

. l'affichage de la valeur affectée à une variable

. l'affectation, à une variable symbolique, d'une constante ou le résultat implicite d'une expression.

Jusque là, c'est élémentaire et cela ne présente guère d'intérêt. Dans l'expression précédente, le symbole + est en APL, une fonction dyadique, c'est-à-dire, une fonction qui agit sur deux variables. Il existe de la même manière, des fonctions niladiques (sans argument) et des fonctions monadiques (avec un seul argument). Chaque fonction APL est représentée par un symbole et, avec l'APL 80 dans sa version cassette, on dispose de la plupart des fonctions APL. L'astuce utilisée pour représenter les symboles spécifiques est très facile à retenir, et se rapproche beaucoup des symboles APL : en appuyant sur la touche "shift", on obtient les lettres, mais précédées d'un point, ainsi la fonction APL L est représentée sur l'écran par .L, de même T par .T etc...

Nous avons vu comment affecter à une variable un nombre scalaire ; de la même manière, on peut affecter à une variable une suite de nombres, qui correspondent aux composantes d'un vecteur :

```
VECTEUR ⟵ 1 2 3 4 5
```

ou une chaîne de caractères :

```
CHAINE ⟵ 'ABCDEF'
```

Une particularité de l'APL est l'emploi de chaînes de caractères vides :

```
CHVIDE ⟵ ''
```

ou de vecteurs vides ; ces derniers sont généralement obtenus par l'application de certaines fonctions. Par exemple, la fonction génératrice d'indices .IN (ιN en APL) donne une suite d'entiers de 1 à N :

```
SUITE ⟵ .I 5
SUITE
1 2 3 4 5
```

Si l'argument est nul, le vecteur est vide

```
V ⟵ .I Ø
V
(vide)
```

On voit donc qu'en APL il n'existe pas de déclarations comme en BASIC ou PASCAL. Les opérations sur les vecteurs, ou plus généralement sur les tableaux à plusieurs dimensions, se font exactement comme sur les scalaires :

```
V1 ⟵ 1 2 3
V2 ⟵ 2 4 6
W  ⟵ V1 + V2
```

En dehors des nombreuses fonctions APL, des opérateurs, associés à une fonction, permettent des formes de fonctions particulièrement intéressantes : par exemple, la "réduction" f/V applique la fonction f sur les éléments du vecteur du tableau V de la manière suivante :

f/V est équivalent à V1 f V2 f V3 si V1, V2, V3 sont les éléments de l'ensemble V ainsi +/1 2 3 est équivalent à 1+2+3.

La somme des 10 premiers nombres devient alors très facile à programmer !

```
S ⟵ +/.I 10
```

Maintenant, on commence à voir nettement l'intérêt de ce langage. L'ensemble des fonctions APL peut être complété par des fonctions créées par l'utilisateur ; celles-ci s'emploient rigoureusement de la même manière que les autres fonctions. En APL 80, pour passer en mode définition, on utilise la commande)DEF, par exemple :

```
)DEF  R ⟵MOY    V
1 : R ⟵(+/V) % .P
```

Pour quitter le mode définition, il faut enfoncer la touche "Break". Cette fonction MOY calcule la moyenne d'une suite de nombre représentée par V. Pour l'appeler, il suffit d'écrire :

```
W ⟵ 1.4   2   2.6   3.2   0.8

MOY   W
```

Remarquez la facilité d'appel d'une fonction : les arguments sont transmis automatiquement, quel que soit leur type.

Dans une "fonction utilisateur", on peut utiliser la 3ème (et dernière) instruction APL : le branchement, représenté par le symbole ⟶ qui signifie aller à l'instruction dont le numéro est donné par l'évaluation de l'expression située à droite du symbole ⟶ . Soit n le résultat de l'expression (qui peut être tout simplement un nombre) ; alors, si n ne correspond à aucun numéro d'instruction de la fonction, on quitte la fonction : c'est l'équivalent de RETURN en Basic ou en Fortran ; si n est un vecteur vide (ou chaîne de caractère vide), l'exécution de la fonction continue en séquence (l'instruction de saut est ignorée). Les deux programmes donnés en exemple assurent le calcul des racines d'un trinôme (les listes données ici ont été obtenues à partir de la version disque d'APL 80). La première version est assez détaillée et fait appel à la fonction IF, définie de la manière suivante :

```
)DEF   Z ⟵ A IF B
1 : Z ⟵ B/A
```

La deuxième version est plus concise.

```
0:    RACINE←TRINOME ABC
1:    A ←ABC ( 1)
2:    B ←ABC ( 2)
3:    C ←ABC ( 3)
4:    DISCRIM ←(B xB )_ 4xA xC
5:    →RACDBLE IF DISCRIM = 0
6:    →RACINES IF DISCRIM ) 0
7:    →ZERORAC IF DISCRIM ( 0
8:    RACDBLE :X ←_B % 2xA
9:    RACINE ←X ,X
10:    → 0
11:    RACINES :D ←DISCRIM * .5
12:    X1 ←(_B _D )% 2xA
13:    X2 ←(_B +D )% 2xA
14:    RACINE ←X1 ,X2
15:    → 0
16:    ZERORAC :'PAS DE RACINE'

TRINOME 1 -1 -6
3 -2
TRINOME 1 4 4
-2 -2
TRINOME 1 1 1
PAS DE RACINE
```

```
0:   R←TRN V
1:   D ←(V ( 2)xV ( 2))_ 4xV ( 1)xV ( 3)
2:   →(L1 ,L2 )( 1+D ( 0)
3:   L1 :D1 ←D * .5
4:   X1 ←(_V ( 2)_D1 )% 2xV ( 1)
5:   X2 ←(_V ( 2)+D1 )% 2xV ( 1)
6:   R ←X1 ,X2
7:   → 0
8:   L2 :'PAS DE RACINE'
```

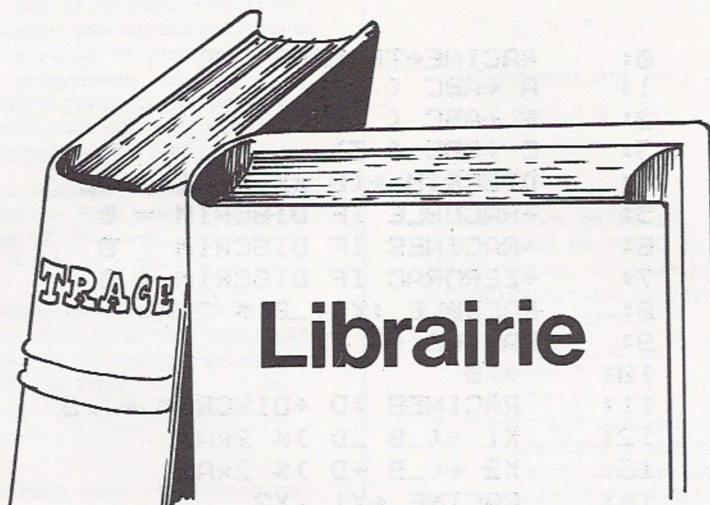
En APL, il n'existe pas d'instructions de boucle, il faut donc initialiser, incrémenter et tester le compteur. Cela n'est pas très gênant, car l'utilisation de certaines fonctions évite l'emploi de boucles, donc leur programmation est moins fréquente qu'avec les autres langages.

Finalement, ce qui est le plus intéressant, c'est de charger sur le TRS-80 (machine à tout faire!), la cassette APL 80 et de créer de nombreux programmes, avec évidemment, au début, des erreurs de programmation. Que le futur utilisateur d'APL soit rassuré, les erreurs sont signalées explicitement et, par la très forte interactivité, la correction est aisée. Au bout de quelque temps, le nouvel APListe aura envie de découvrir APL 80-disque, qui, pour une dépense encore modeste, donne presque toutes les possibilités de l'APL et permet également d'utiliser des fonctions écrites en langage machine.

On a tendance trop fréquemment à démontrer que tel langage est plus intéressant qu'un autre, les comparaisons sont toujours délicates et à mon avis, il n'est pas possible d'affirmer qu'un langage est dans tous les cas meilleur qu'un autre.

Chaque langage de programmation possède ses avantages et ses inconvénients et surtout convient mieux à certains types de traitement. Pour avoir une bonne opinion, il faut pratiquer : le TRS-80 est une machine universelle qui autorise l'emploi de nombreux langages (assembleur, Fortran, APL, Pascal, Forth, Lisp...). A chacun ses préférences !

Claude NOWAKOWSKI



THE CUSTOM TRS-80 & OTHER MYSTERIES
Par Dennis Kitsz
Editeur : IJG Inc.
Prix : 280 F TTC
335 pages - broché

Si vous êtes de ceux qui cherchent à transformer leur modèle 1 par des modifications du logiciel ou du matériel en une machine plus puissante, ce livre est pour vous. De même si vous voulez simplement mieux connaître comment marche votre ordinateur, vous trouverez dans cet ouvrage une mine de renseignements.

Il y a tout de même une réserve à faire en ce qui concerne la primeur de certains chapitres. Dennis Bathory Kitsz qui signe ce livre est un des auteurs américains les plus prolifiques dans le domaine des extensions du TRS-80. Ses articles paraissent dans plusieurs revues américaines spécialisées dans ce matériel. Le noyau de ce livre est une compilation d'articles publiés précédemment. Si vous lisez déjà tout ce qui s'écrit sur votre machine, vous serez peut-être un peu déçus de ne pas découvrir grand chose de nouveau.

A moins que, comme moi, vous n'ayez quelques difficultés à retrouver l'article que vous cherchez dans des piles de revues qui grandissent chaque mois :

est-ce la pile du bureau, celle du coin repassage, celle du garage ? A moins que dans les cartons à la cave ... ? Ah, si on avait un peu de temps, on pourrait écrire un programme qui ...! Enfin c'est assez pratique d'avoir réuni toutes ces idées dans un volume suffisamment épais pour qu'on ne l'égare pas facilement.

Il faut dire que ce livre est tout de même augmenté de plusieurs chapitres qui fournissent des informations de base pour vous mettre en mesure d'attaquer avec quelques chances de succès les modifications matérielles proposées.
Vous y trouverez la liste des outils et du petit matériel courant qui sont nécessaires, une introduction à la théorie de l'électronique, des indications sur la manière de lire et d'interpréter un schéma, des conseils pratiques pour souder, manipuler des circuits ou simplement monter ou démonter l'ordinateur ou l'interface. J'ai trouvé ici un conseil qui m'a été tout à fait utile. Depuis quelque temps, à force de modifications et d'ajouts, le clavier de mon modèle 1 baillait lamentablement : plus moyen de le fermer complètement. C'était un vrai plat de spaghetti ! Grâce au conseil tout simple de ce livre, tout tient à nouveau, et les vis sont remises en place.

Une autre partie du livre, à ma connaissance inédite, se rapporte aux pannes de l'ordinateur, de ses périphériques les plus courants, la manière de les éviter, de les diagnostiquer et de les réparer.

Nous en venons à la partie essentielle, qui justifie le titre du livre, "le TRS-80 sur mesure". Des modifications de logiciel d'abord, routines de driver (ou "pilotes" - parlons français, bien que ce livre soit, lui, en anglais !). Pilotes, disions nous donc, de clavier (antirebond, touches à répétition, minuscules, graphiques, sonores, etc) d'imprimante, de lecture rapide des cassettes. Des extensions du BASIC, des chargeurs spéciaux de programmes, l'autoexécution d'un programme BASIC charge à partir d'une cassette, programmes de musique, etc, etc .

Un nombre assez impressionnant de modifications du matériel sont ensuite proposées. Nous trouvons, à ce sujet : le branchement d'un deuxième clavier, l'ajout d'un clavier hexadécimal, la vidéo inverse, totale ou programmable caractère par caractère, les minuscules, la vitesse rapide, les MEM niveau 1 et 2 dans la même machine, les extensions de mémoire, le graphique haute résolution, le remplacement de la MEM par de la MEV, l'interfaçage avec des périphériques non standard, l'horloge temps réel, l'utilisation de la partie de mémoire non occupée pour créer des "bancs" de mémoire sélectables et extensibles presque à l'infini ... et d'autres encore que je vous laisse découvrir par vous-mêmes.

Certaines modifications paraissent très complexes, et demandent certainement une solide aptitude manuelle et théorique. Mais même sans aller jusqu'à la réalisation systématique des montages proposés, ce livre nous apprend beaucoup sur le modèle 1 et ses périphériques, et mérite sa place dans votre bibliothèque.

LE TOUR DU TRS-80 EN TROIS TOMES, avec :
La pratique du TRS-80, tomes I, II et III
Par Pierre Giraud et Alain Pinaud
Editions du PSI
Prix respectifs : 65, 85 et 65 F TTC
128, 212 et 128 pages - brochés

Existe-t-il des propriétaires de modèle 1, de machines compatibles ou même de modèle 3 qui n'ont pas lu au moins le premier tome de cette série ?

J'en connais un certain nombre qui ont commencé par ces livres pour mieux arrêter leur choix ou pour se laisser persuader que leur vie sans ordinateur individuel serait désormais impossible !

Voilà, en tout cas, une série utile pour faire connaissance - en français - avec la machine que vous envisagez d'acheter, que vous venez d'acheter, ou que vous voulez utiliser au mieux.

Le tome 1 présente l'ordinateur, ses périphériques et ses logiciels de base. Il donne une introduction suffisante de la théorie du matériel (logique élémentaire, microprocesseur) et du logiciel (calcul et logique binaire, commandes d'édition, principes de base de la programmation) pour pouvoir aborder ensuite la présentation du BASIC. Celui-ci n'est pas présenté instruction par instruction mais plutôt selon une progression qui facilite sa compréhension et son apprentissage. Cette description du BASIC est étayée par des exemples pratiques qui montrent l'action de chaque instruction ou fonction du langage. A la suite, plusieurs petits programmes bien documentés et expliqués font la synthèse des chapitres précédents. Puis ce livre se termine sur un résumé des instructions BASIC avec leurs fonctions et diverses tables de conversion. A noter également qu'une édition récente de cet ouvrage consacre au modèle 3 un chapitre entier.

Le tome 2 aborde un sujet plus difficile : si nous programmions en assembleur ?

Ce tome apporte de nombreuses informations pour programmer en assembleur sur le TRS-80, mais ce n'est hélas pas une initiation et il est supposé que le lecteur possède déjà les rudiments du langage.

On y aborde également quelques instructions BASIC surtout utilisées en relation avec l'assembleur comme PEEK, POKE, INP, OUT, DEF FN, USR, DEF USR, etc .

Y figurent aussi des informations pratiques sur l'utilisation de l'éditeur/assembleur EDTASM, et sur quelques programmes d'aide à la mise au point. Les exemples d'application (extensions BASIC, tracé graphique d'un segment de droite, par exemple), sont un bon début pour la construction de notre bibliothèque de routines assembleur.

Sachant qu'il y a un manuel avec le modèle 1 comme il y a un manuel avec l'éditeur-assembleur, on peut se demander si ces deux livres ne font pas double emploi avec les manuels standard.
En réalité, il n'y a pas deux auteurs qui disent la même chose avec les mêmes mots, et pour apprendre un sujet nouveau, lire et consulter plusieurs ouvrages sur le même sujet (appelons cela la lecture parallèle) est une des méthodes les plus efficaces qui soient. Sans compter que ces livres apportent en français des informations de base introuvables dans les manuels d'utilisation.
Deux petits reproches toutefois, sur cet ouvrage : les premières éditions comportaient un certain nombre d'erreurs propre à plonger le lecteur novice dans la plus profonde confusion. Il semble, fort heureusement, que cela ne soit plus le cas maintenant. Vérifiez donc, si vous achetez cet ouvrage, qu'il s'agit bien d'une édition récente (la date figure en fin de livre). Je n'ai pas compris, d'autre part, le malin plaisir que les auteurs prennent à décrire un programme de mise au point - paraît-il "efficace" - mais que l'on ne peut trouver ni acheter nulle part !

Mais venons-en au tome 3 :
vous pouvez préparer votre fer à souder ! Ou peut être voulez-vous simplement savoir et comprendre ce qui se cache sous votre clavier, ou dans votre interface d'extension ?
Première lecture : c'est si compliqué que çà ?
Deuxième lecture : ah oui, c'est bien sûr !
Troisième lecture : finalement ce n'est pas si difficile !

Un livre qu'on peut lire trois fois au moins, et même plus, pour le prix d'un seul, cela ne se trouve pas partout.
Plus sérieusement, disons que ce livre est d'un abord parfois difficile pour un non-spécialiste. Je ne sais si le sujet pouvait être traité de façon plus abordable. Il m'a vraiment fallu "m'accrocher" pour le comprendre, et je ne l'ai fait que parce que le sujet m'intéressait beaucoup : il vous permet sur votre modèle 1 d'augmenter la vitesse, d'installer un "beep" sonore, des minuscules, et un bouton reset accessible et qui marche en toutes circonstances, tout cela semble donc une bonne affaire. Mais attention tout de même : ce livre est dangereux ... et ne doit pas tomber entre des mains innocentes !

Je profite de cet article pour lancer un appel aux copains à qui j'ai prêté ces livres. En fait, je les ai prêtés souvent, et je ne sais plus qui les a en ce moment ! Si vous pouviez me les rapporter, cela m'arrangerait beaucoup.

En attendant il ne me reste qu'à espérer la sortie d'un tome 4, car les disques, le DOS et le BASIC disque, qui mériteraient bien un traitement identique, n'ont pas été abordés dans les trois tomes précédents.

Wolfgang Lauter

Trucs à Brac

DEUX TOUCHES SHIFT, COMME TOUT LE MONDE !

Le modèle 3, comme la plupart des ordinateurs, possède deux touches SHIFT sur son clavier. Rien de surprenant jusque là, direz-vous. Mais saviez-vous qu'il est possible de distinguer par programme, la touche gauche de la touche droite ? Cela fait partie des petits détails qui distinguent le modèle 3 du modèle 1 et cela s'en ressent sur certains programmes en langage machine qui ignorent tout simplement la touche SHIFT de droite. Sur le modèle 1, les deux touches SHIFT étaient physiquement reliées ensemble, et la lecture de leur état pouvait se faire en consultant l'adresse 3880H : 1 signifiait que l'une des touches SHIFT était enfoncée. Sur le modèle 3, les deux touches sont séparées, et la consultation de l'adresse 3880H donne cette fois :

- 0 : aucune touche SHIFT enfoncée
- 1 : touche SHIFT gauche enfoncée
- 2 : touche SHIFT droite enfoncée
- 3 : les deux touches enfoncées

PEEK/POKE SUR ENTIERS SEULEMENT

Les instructions POKE et DEFUSR (version disquette), ainsi que les fonctions PEEK et VARPTR, opèrent uniquement sur des valeurs entières, c'est-à-dire comprises entre -32768 et +32767. Cela signifie que lorsque l'on veut atteindre l'adresse mémoire 32768, il faut utiliser la valeur -32768 ... La fin de mémoire (OFFFFH) correspondra à la valeur -1. Bien entendu, le problème ne se pose pas pour les utilisateurs de configuration 16K, la dernière adresse de mémoire vive étant justement 32767. Pour les autres utilisateurs (équipes de l'extension mémoire), la conversion peut se faire ainsi (AD étant l'adresse mémoire a atteindre) :

```
IF AD > 32767 THEN AD=1*(65536-AD)
POKE AD,...
```

Mais si l'on désire une formule plus "linéaire" et qui, de plus, évite un test de condition, on peut tout simplement faire :

```
AD=AD-INT(AD/32768)*65536
POKE AD,...
```

En appliquant cette formule une seconde fois, on retrouve la valeur initiale ... Méditez la-dessus !

QUESTION DE PILOTAGE ...

Certains programmes en langage machine commencent leur initialisation en restituant dans les blocs de contrôle concernés, l'adresse ROM des Pilotes (drivers) clavier et écran. Sur le modèle 1, cela a pour effet de perdre l'éventuel traitement anti-rebonds du clavier chargé par le DOS notamment. Sur le modèle 3, les mêmes programmes ne fonctionnent plus du tout ! Cette dernière cause est simple (mais le remède aussi) : les adresses des pilotes clavier et écran sont différentes entre les deux modèles :

	Adresse pilote clavier dans bloc de contrôle		Adresse pilote écran dans bloc de contrôle	
	4016H	4017H	401EH	401FH
Mod 1	E3H	03H	58H	04H
Mod 3	24H	30H	73H	04H

Avec un utilitaire de recherche, il vous suffira de remplacer les "E3 03" par des "24 30" et les "58 04" par des "73 04". En somme, rien de bien méchant !

DES "OU" ET DES "ET" SUR POQUETTE

Sur le poquette, il n'est pas possible, à l'instar du modèle 1, de faire :

```
IF A > 1 AND B > 2 ...
ou bien :
IF A > 1 OR B > 2 ...
```

Par contre, on peut produire le même effet en écrivant :

```
IF (A > 1)*(B > 2) ...
ou bien :
IF (A > 1)+(B > 2) ...
```

comme cela était aussi le cas sur le "level 1" du modèle 1.

TRUC DANS LA POCHE !

Lorsque vous voulez imprimer le résultat d'un calcul sur l'imprimante (en mode calcul), vous devez passer en mode "PRO", puis faire : 10 PRINT (valeur à calculer). Cette solution n'est pas très satisfaisante et je lui préfère celle-ci :

```
10:"C"AREAD A
20:PRINT A
25:PRINT " ":
   PRINT " ":
   PRINT " "
30:END
```

Il suffit alors de frapper l'expression à calculer en mode DEF, et d'appuyer sur SHFT C pour imprimer le résultat.

(Jean-David Blanc)

BACKUP REJETE ?

Rien de plus énervant, en voulant faire une sauvegarde d'une disquette TRSDOS, de se faire refuser le "backup" sous prétexte que la disquette destination contient déjà des informations. Il y a deux solutions à ce problème : effacer la disquette à l'aide d'un aimant que l'on "ballade" sur la pochette (demande certaines précautions !) ou formatter préalablement la disquette (avec l'utilitaire FORMAT) en donnant à la disquette le MEME NOM que celle que l'on veut dupliquer, ainsi que le MEME MOT DE PASSE (Password) ... si l'on se souvient de celui-ci ! Le programme FORMAT détecte alors que la disquette contient des informations, et il suffit de répondre "Y" (yes) à la question. Le BACKUP se passe alors sans problème.

utilitaire

Un programme pour mettre à jour vos disquettes

Amateurs de cassettes, tournez la page (mais il faudra tout de même y penser un jour, non ?)! Ce programme ne concerne que les possesseurs de minidisquettes utilisant un système d'exploitation capable d'exécuter l'ordre COPY sous BASIC (LDOS, Newdos ou Newdos80 par exemple).
Prenons le cas d'Apparat, créateur du Newdos, qui publie régulièrement des "zaps" (modifications) de ses systèmes d'exploitation. Cela part d'un excellent principe d'assurer la "maintenance" de ses produits, mais cela devient une véritable corvée lorsqu'il faut appliquer ces modifications sur toutes ses disquettes. Avec le programme ce-dessous, il suffit d'effectuer les corrections concernées sur une seule disquette (que nous appellerons disquette mère), et la copie sur les autres disquettes se fera automatiquement et de manière sélective. Cela signifie que le fichier SYS10/SYS par exemple, ne sera copié sur la disquette de destination (celle devant être mise à jour) que si son ancienne version y figure déjà.

Un fichier descripteur constitué par Scripsit par exemple, décrira le nom des programmes devant être modifiés. Dans le cas de Newdos-80, nous l'avons appelé ZAP54/DAT (54 étant le numéro du dernier "zap") et il est constitué ainsi :

```
SYS0/SYS
SYS2/SYS
SYS3/SYS
SYS6/SYS
SYS8/SYS
SYS11/SYS
SYS12/SYS
SYS14/SYS
SYS16/SYS
SYS17/SYS
SYS18/SYS
BASIC/CMD
EDTASM/CMD
DISASSEM/CMD
SUPERZAP/CMD
```

En lancant MAJZAP, tous les programmes de cette liste figurant aussi sur la disquette de destination seront mis à jour.
Normalement, le programme tel qu'il est décrit est utilisable pour des configurations de deux minidisquettes mais peut être adapté à un ensemble monodisquette en modifiant la chaîne A$ située en ligne 200.

```
10 ' Mise a jour des "zaps" - (C) TRACE et JP du Tillet       ' F6
20 '                                                          ' EC
30 CLS : PRINT,"M I S E   A   J O U R   D E S   Z A P S"     ' 39
40 PRINT,STRING$(39,CHR$(131))                                ' 70
50 CLEAR 1000 : DIM NF$(50)                                   ' 23
60 LINEINPUT"-NOM DU FICHIER ZAP ?" ; NZ$                     ' 80
70 INPUT"NUMERO DISQUE SOURCE (O A 4)";SD$ : SD$=":"+SD$      ' BF
80 INPUT"NUMERO DISQUE DESTINATION (O A 4)";DD$:DD$=":"+DD$   ' 71
90 N=1 : OPEN "I",1,NZ$                                       ' 94
100 IF EOF(1) THEN 140                                        ' 43
110 LINEINPUT#1,NF$(N)                                        ' 96
120 IF NF$(N)="" OR LEFT$(NF$(N),1)=" " THEN 140              ' 41
130 N=N+1 : GOTO 100                                          ' B7
140 N=N-1                                                     ' 04
150 CLOSE : IF N=0 THEN CLOSE : PRINT "FICHIER VIDE...":END   ' A2
160 ON ERROR GOTO 250                                         ' FD
170 FOR X=1 TO N                                              ' 6C
180 PRINT " -FICHIER ";NF$(X);TAB(36);                        ' CF
190 OPEN "I",1,NF$(X)+DD$                                     ' B0
200 CLOSE : A$="COPY "+NF$(X)+SD$+" TO "+DD$                  ' BC
210 CMD A$ : PRINT "---> MIS A JOUR SUR ";DD$                 ' 04
220 NEXT X                                                    ' 45
230 PRINT : PRINT "DISQUETTE ";DD$;" MISE A JOUR."            ' 3D
240 ON ERROR GOTO 0 : END                                     ' 5A
250 CLOSE : IF ERR<>108 AND ERR <>106 THEN ON ERROR GOTO 0    ' F5
260 PRINT "   (NON TROUVE SUR ";DD$;")" : RESUME 220          ' 17
```




```
              M I S E   A   J O U R   D E S   Z A P S

 -NOM DU FICHIER ZAP ? zap54/dat
 NUMERO DISQUE SOURCE (O A 4)? 0
 NUMERO DISQUE DESTINATION (O A 4)? 2
  -FICHIER SYS6/SYS                     ---> MIS A JOUR SUR :2
  -FICHIER SYS16/SYS                       (NON TROUVE SUR :2)
  -FICHIER SUPERZAP/CMD                    (NON TROUVE SUR :2)
  -FICHIER SYS0/SYS                     ---> MIS A JOUR SUR :2
  -FICHIER SYS14/SYS                       (NON TROUVE SUR :2)
  -FICHIER BASIC/CMD                       (NON TROUVE SUR :2)
  -FICHIER EDTASM/CMD                      (NON TROUVE SUR :2)
  -FICHIER DISASSEM/CMD                    (NON TROUVE SUR :2)
  -FICHIER SYS3/SYS                     ---> MIS A JOUR SUR :2
  -FICHIER SYS2/SYS                     ---> MIS A JOUR SUR :2
  -FICHIER SYS11/SYS                       (NON TROUVE SUR :2)
  -FICHIER SYS17/SYS                       (NON TROUVE SUR :2)
  -FICHIER SYS12/SYS                       (NON TROUVE SUR :2)
  -FICHIER SYS18/SYS                       (NON TROUVE SUR :2)
  -FICHIER SYS8/SYS                     ---> MIS A JOUR SUR :2

 DISQUETTE :2 MISE A JOUR.
```

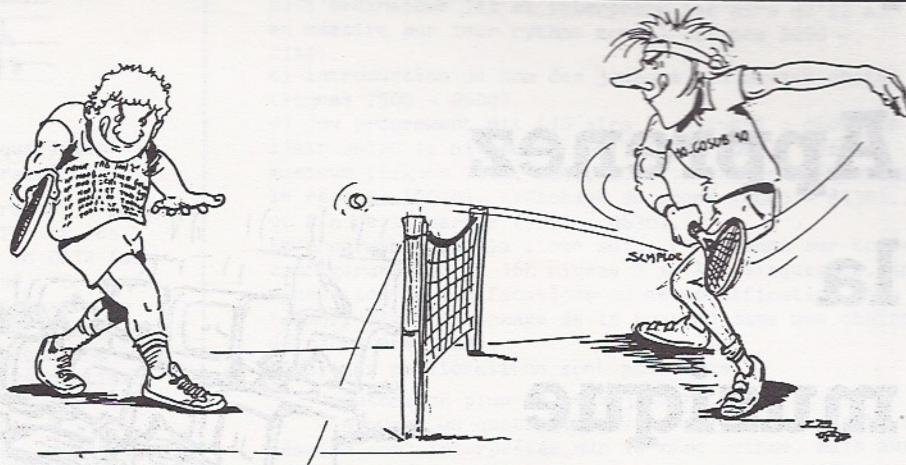
Pour ceux qui ont la chance de posséder la version 2 de NEWDOS-80, ce programme est superflu car un nouveau paramètre de la commande copié (DFO = Destination File Only) permet justement de copier uniquement les programmes figurant à la fois dans le fichier descripteur ET sur la disquette de destination. La commande est la suivante :

```
COPY,0,1,,NFMT,CBF,ILF=ZAP54/DAT,DFO
```

pour une disquette mère en 0 et une disquette destination en 1.

Jean-Pierre du Tillet

Côté court



Cette rubrique contient des programmes écrits en BASIC, mais dont l'originalité est que la taille ne dépasse pas TROIS lignes au maximum, chacun d'eux effectuant une fonction bien précise. Creusez-vous les méninges et soumettez-nous les vôtres !

- TEST MEMOIRE : UNE LIGNE (si ! si !)

```
10 GOSUB 10
```

Si vous avez le moindre doute sur votre mémoire ou sur l'intégrité de votre système (et que vous arriviez à charger le BASIC), essayez cette ligne de programme : elle vaut bien des tests mémoire sophistiqués car elle met en oeuvre le système de pile de l'interprêteur. Si tout se passe bien, le message "OUT OF MEMORY" doit s'afficher au bout de quelques instants, dépendant de la taille mémoire de votre système. Dans tout autre cas, il y a effectivement un problème ... Pour continuer toute autre opération, faire CLEAR. (Programme communiqué par Jacques Eeses).

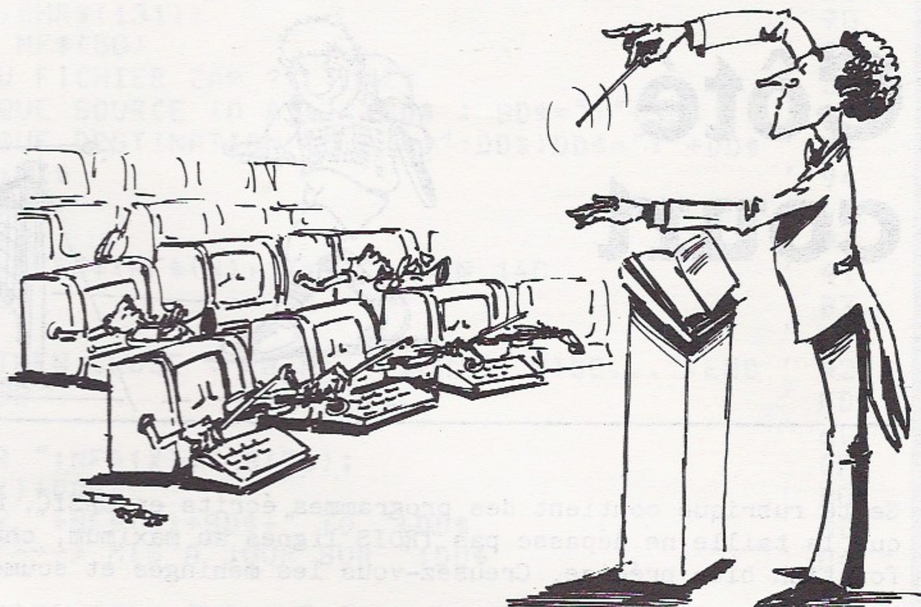
- CONVERSION HEXADECIMAL --- DECIMAL : 2 LIGNES.

```
1000 H$="0123456789ABÇDEF" : D=0 : M=1 : FOR L=1 TO LEN(N$) : FOR I=0 TO 15 : IF
MID$(H$,I+1,1)<>MID$(N$,LEN(N$)-L+1,1) THEN NEXT
1010 D=D+I*M : M=M*16 : NEXTL : RETURN
```

Cette routine de conversion est appelée par GOSUB. En entrée, la variable N$ contient la valeur hexadécimale (sans & ou H), en sortie, la variable D contient la valeur décimale. La chaîne H$ peut être extérieure à la routine.
Exemple d'utilisation :

```
10 INPUT "VALEUR HEXA" ; N$
20 GOSUB 1000
30 PRINT "VALEUR DECIMALE :" ; D
```

Apprenez la musique



Vous connaissez certainement les jeux radiophoniques consistant à trouver le plus rapidement possible, certains morceaux de musique. Nous vous proposons de vous entraîner à de tels jeux, en vous amusant... tout en utilisant des possibilités supplémentaires offertes par votre modèle 1 : il peut, bien sûr, jouer des airs connus, mais aussi changer de rythme ou même improviser !!

Il ne faut tout de même pas être trop ambitieux : un système muni d'un amplificateur téléphonique ne se compare pas au "Philharmonique" de Berlin sous la baguette de Karajan, ou encore aux synthétiseurs de style "Orchestra 80" (voir TRACE n°1) qui offrent 4 voies simultanées, différents timbres d'instruments etc...

Je n'expliquerai pas comment un sous-programme en langage machine permet d'obtenir les notes de plusieurs gammes : un de mes homonymes l'a déjà expliqué dans les numéros 29 et 30 de l'Ordinateur Individuel. Ses idées sont reprises dans le sous-programme musical de ce jeu (lignes 1500 à 1670). Les quelques modifications apportées sont nécessitées par le stockage d'airs de musique, qu'il faut coder de façon simple et peu encombrante en mémoire.

En bref, le son émis dépend de deux paramètres :

1) sa"durée", notée DU ou V
2) sa "hauteur" notée HT, qui est passée dans le registre HL du microprocesseur, par l'intermédiaire de la fonction USR.

Dans l'article mentionné plus haut, la durée était déterminée par sa valeur V selon le codage :

V	0	0,5	1	1,5
note	quadruple croche	quadruple croche pointée	triple croche	triple croche pointée

Il m'a paru plus simple de coder la durée d'une note sur un entier de 0 à 9, ce qui n'occupera qu'un seul octet en mémoire (et non pas 3 pour la valeur 1,5). Ainsi, on peut faire varier la durée de la triple croche (0) à la blanche pointée (9), ce qui est suffisant pour le jeu proposé qui utilise le codage suivant :

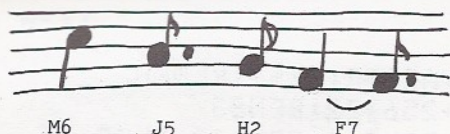
0 = triple croche	1 = triple croche pointée
2 = double croche	3 = double croche pointée
8 = blanche	9 = blanche pointée

En ce qui concerne la hauteur, l'article de l'Ordinateur Individuel utilisait deux paramètres : NG (numéro de la gamme de 0 à 9) et NT (nombre de demi-tons depuis le DO de la gamme considérée, de 0 à 11). Là aussi, il m'a semblé plus simple de n'utiliser qu'un seul paramètre (une lettre) selon le code :

	gamme n°0				gamme n°1				gamme n°2	
note :	do	do	re...si		do	do	re...si		do	do
code :	A	B	C	L	M	N	O	X	Y	Z
ASCII :	65	66	67							90

Apparemment, nous sommes limités à deux gammes, ce qui, une fois encore, suffit pour le jeu proposé. Comme ce qui importe est le nombre de demi-tons séparant une note du DO de départ (et que ce nombre est le code ASCII de la lettre), on peut continuer au-delà du Z (do de la gamme n° 2) par les caractères codés 91 (flèche haute), 92 (flèche basse)... Il faudra alors modifier le tableau des hauteurs HT dans le sous-programme musical.

Pour résumer et illustrer ce propos, le début de "trois jeunes tambours" est codé :



Le tout sera placé dans une chaîne, ce qui limite la durée de l'air à 127 notes (soit 254 caractères).

Certains airs sont stockés dans le programme, à partir de la ligne 11000. Les titres (tableau T$) et les codes des airs (tableau A$) sont placés en DATA (s/p de lecture des airs : lignes 2000 à 2140). Ce système m'a semblé plus simple, pour les utilisateurs de magnétophone, que celui d'un fichier d'airs sur cassette, lu par INPUT -1. Une fois le codage d'un air assimilé, vous pourrez évidemment personnaliser ce jeu en y incluant vos airs favoris.

Faire jouer un air stocké est très simple : par une boucle, on extrait le code de la durée (VAL(MID$(A$( ), ,1))) et le code de la hauteur (ASC(MID$(A$( ), ,1))-65. Il suffit d'appeler la durée et la hauteur stockées dans deux tableaux par DU (VAL(MID$(A$( )...) et HT (ASC(MID$(A$( )...). C'est plus long à écrire qu'à comprendre... et il vaut mieux consulter les lignes 7000 à 7640 !

Tel qu'il est rédigé, le jeu se déroule ainsi :

a) présentation (lignes 1000 à 1140) avec ses différents niveaux :
1 = l'ordinateur ne joue que les airs stockés en mémoire.
2 = airs stockés avec quelques improvisations.
3 = airs stockés interprétés sur un autre rythme avec improvisations.

b) l'ordinateur lit et interprète les airs qu'il a en mémoire sur leur rythme normal (lignes 2000 - 2140).
c) introduction du nom des joueurs, du niveau choisi (lignes 2500 - 2600).
d) jeu proprement dit (10 airs à deviner) : choix de l'air selon le niveau de jeu (lignes 5000 - 5260), musique (lignes 7000, 7640, 9000), test de l'éventuelle réponse (9010), affichage du score (8000 - 8130)... et fin de la partie (9500 - 9630).

Le programme dont la liste suit, fonctionne sur toutes configurations, du 16K niveau 2 au 48K disquettes, sans nécessiter de modifications ou de spécification d'un "memory size" (stockage de la routine dans une chaîne de caractères ZZ$).

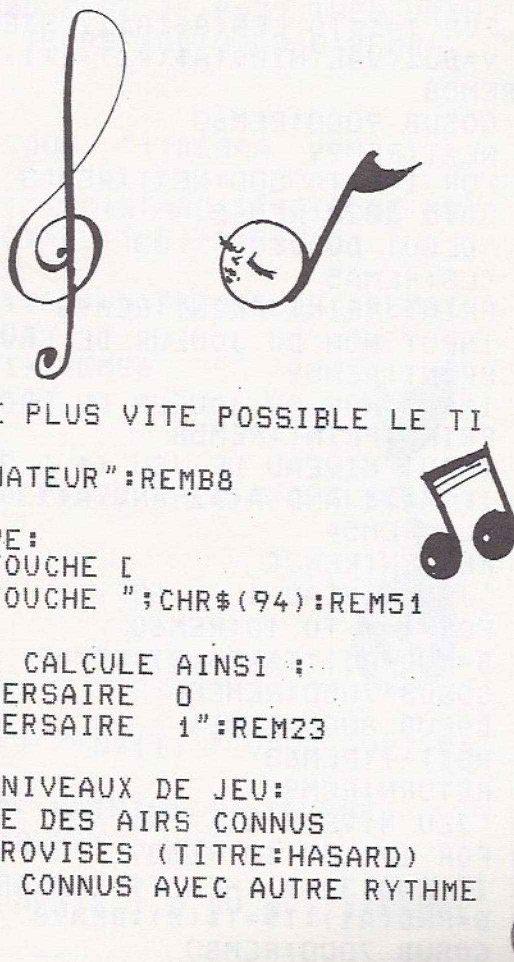
Plusieurs améliorations sont possibles :

- autoriser plus de deux joueurs,
- ajouter un quatrième niveau de jeu : les airs en mémoire sont interprétés sur le même rythme, mais sur une autre mélodie (peut-on encore parler du "même air" ?). J'ai tenté de jouer à ce niveau et j'ai dû abandonner : impossible de trouver quel air est joué ! Mais peut-être avez-vous une oreille plus fine que la mienne !
- essayer de deviner les airs en un nombre de notes fixé à l'avance,
- modifier le sous-programme en langage machine pour qu'il teste si un joueur a trouvé l'air (ce test n'est réalisé en BASIC qu'une fois la note jouée, ce qui n'est pas très équitable)...

Bon amusement !

Cyril Grandpierre

```
5 ' PROGRAMME MUSICAL - (C) TRACE et Cyril Grandpierre   FB
10 CLEAR 1000:REM7D
50 GOSUB 1000:REM7C
60 GOSUB 1500:REM6D
70 GOSUB 2000:REM67
80 GOSUB 2500:REM58
90 ON A GOSUB 5000,5100,5200:REM89
100 GOTO 9500:REM41
1000 'PRESENTATION DU JEU '   15
1010 CLS:REM87
1020 PRINTCHR$(23):PRINT:PRINT:REMCA
1030 PRINTTAB(8)"JEU MUSICAL":REMF1
1040 FOR I=1 TO 2000:NEXT:REMDC
1050 CLS:REM5E
1060 PRINT"CE JEU CONSISTE A TROUVER LE PLUS VITE POSSIBLE LE TI
TRE
D'UN AIR MUSICAL INTERPRETE PAR L'ORDINATEUR":REMB8
1070 PRINT:REM1C
1080 PRINT"QUAND IL PENSE L'AVOIR TROUVE:
    LE JOUEUR DE GAUCHE APPUIE SUR LA TOUCHE [
    LE JOUEUR DE DROITE APPUIE SUR LA TOUCHE ";CHR$(94):REM51
1090 PRINT:REM08
1100 PRINT"SELON LA REPONSE LE SCORE SE CALCULE AINSI :
    REPONSE CORRECTE :  JOUEUR  2   ADVERSAIRE  0
    REPONSE FAUSSE   :  JOUEUR  0   ADVERSAIRE  1":REM23
1110 PRINT:REMF4
1120 PRINT"VOUS POUVEZ CHOISIR ENTRE 3 NIVEAUX DE JEU:
    1)FACILE    :L'ORDINATEUR NE JOUE QUE DES AIRS CONNUS
    2)MOYEN     :AIRS CONNUS OU AIRS IMPROVISES (TITRE:HASARD)
    3)DIFFICILE:AIRS IMPROVISES OU AIRS CONNUS AVEC AUTRE RYTHME
";:REM11
1130 FOR I=1 TO 5000:NEXT:REM7F
```



```
1140 RETURN:REMF6
1500 'IMPLANTATION PROGRAMME MUSICAL '   1F
1510 DIM DU%(9),HT%(25):REM9E
1520 ZZ$="PLACE POUR ROUTINE SONORE" : AZ=VARPTR(ZZ$):REMED
1530 Z1=PEEK(AZ+1) : Z2=PEEK(AZ+2) : AD=Z1+256*Z2:REM83
1535 IF PEEK(16396)=201 THEN POKE 16526,Z1:POKE 16527,Z2 ELSE DE
FUSR=AD : CMD"T":REMF7
1540 FOR I=AD TO AD+23:REM5E
1550 READ D:POKE I,D:REM79
1560 NEXT I:REM12
1570 AD=AD+5:REMF7
1580 FOR I=0 TO 9:REM09
1590 DU=INT(2↑(I/2+1)+.5):DU%(I)=DU:REM60
1600 NEXT:REM33
1610 FOR I=0 TO 25:REMBD
1620 F=133*(2↑(I/12)):REM04
1630 L=INT((1.774E6/F-70)/52+.5):IF L>255 THEN L=255:REMC7
1640 IH=INT(1275/L+.5):IF IH>127 THEN IH=127:REM56
1650 HT%(I)=256*IH+L:REM42
1660 NEXT:REMF7
1670 RETURN:REME2
2000 'LECTURE DES AIRS '  29
2010 DIM A$(20),T$(20):REM26
2020 CLS:REM91
2030 PRINT:PRINT:PRINT"JE PEUX VOUS INTERPRETER :":REM91
2040 READ T$:REMFE
2050 IF T$="FIN" THEN RETURN:REM9D
2060 A1=A1+1:T$(A1)=T$:REM73
2070 PRINT:PRINT:PRINTT$:REME0
2080 READ A$(A1):REM25
2090 FOR I=1 TO LEN(A$(A1)) STEP 2:REMD7
2100 V=DU%(VAL(MID$(A$(A1),I+1,1))):C=HT%(ASC(MID$(A$(A1),I,1))-
65):REMD8
2110 GOSUB 9000:REM60
2120 NEXT:REM29
2130 FOR I=1 TO 500:NEXT:REMC3
2140 GOTO 2020:REM4B
2500 'DEBUT DU JEU '   33
2510 CLS:REMA5
2520 PRINT:PRINT:PRINT:REM95
2530 INPUT"NOM DU JOUEUR DE GAUCHE ";N$(0):REM87
2540 PRINT:REM59
2550 INPUT"NOM DU JOUEUR DE DROITE ";N$(1):REM58
2560 PRINT:PRINT:REM58
2570 INPUT"NIVEAU DE JEU (1 A 3) ";A:REM78
2580 IF A<>1 AND A<>2 AND A<>3 THEN CLS:GOTO 2560:REM76
2590 CLS:REM54
2600 RETURN:REM3C
5000 'JEU NIVEAU 1 '  65
5010 FOR I=1 TO 10:REM6D
5020 B=RND(A1):T$=T$(B):REM07
5030 GOSUB 7000:REMEF
5040 GOSUB 8000:REME4
5050 NEXT I:REM63
5060 RETURN:REM97
5100 'JEU NIVEAU 2 '  01
5110 FOR I=1 TO 10:REM09
5120 IF RND(3)=1 THEN T$="HASARD":GOSUB 7300:GOTO 5150:REMC0
5130 B=RND(A1):T$=T$(B):REM98
5140 GOSUB 7000:REM80
```

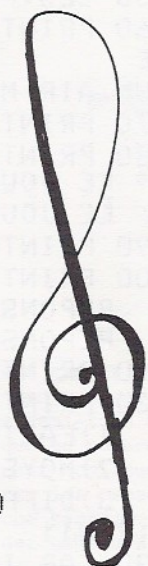












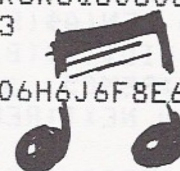
```
5150 GOSUB 8000:REM75
5160 NEXT I:REMF4
5170 RETURN:REM28
5200 'JEU NIVEAU 3 '   9C
5210 FOR I=1 TO 10:REMA4
5220 IF RND(3)=1 THEN T$="HASARD":GOSUB 7300:GOTO 5250:REM5B
5230 B=RND(A1):T$=T$(B):REM34
5240 GOSUB 7600:REM16
5250 GOSUB 8000:REM11
5260 NEXT I:REM90
5270 RETURN:REMC4
7000 FOR J=1 TO LEN(A$(B)) STEP 2:REMC5
7010 V=DU%(VAL(MID$(A$(B),J+1,1))):C=HT%(ASC(MID$(A$(B),J,1))-65
):REMF5
7020 GOSUB 9000:REM1F
7030 IF F=0 THEN NEXT:REM44
7040 RETURN:REMD3
7300 FOR J=1 TO 20+2*RND(10) STEP 2:REM14
7310 C=RND(24):V=DU%(RND(8)-1):REMCF
7320 GOSUB 9000:REMF2
7330 IF F=0 THEN NEXT:REM17
7340 RETURN:REMA6
7600 FOR J=1 TO LEN(A$(B)) STEP 2:REM6B
7610 C=HT%(ASC(MID$(A$(B),J,1))-65):V=DU%(RND(8)-1):REM8F
7620 GOSUB 9000:REMC5
7630 IF F=0 THEN NEXT:REMEA
7640 RETURN:REM79
8000 'SCORE ET AFFICHAGE '   A1
8010 CLS:REM13
8020 IF P=0 THEN PRINT:PRINT:PRINT"PERSONNE N'A REPONDU":PRINT:P
RINT:PRINT"LE REPONSE EST :";T$:GOTO 8080 ELSE IF P=72 THEN PRIN
T:PRINT:PRINT"VOUS AVEZ REPONDU EN MEME TEMPS":GOTO 8080:REMB4
8030 IF P=8 THEN G=0 ELSE G=1:REM65
8040 PRINT:PRINT:REMDB
8050 PRINTN$(G);" A QUEL TITRE PENSEZ- VOUS  ";:REMAA
8060 INPUT R$:REM66
8070 IF R$=T$ THEN PRINT:PRINT"BRAVO !!!":SC(G)=SC(G)+2 ELSE PRI
NT:PRINT"NON,LA REPONSE EST :";T$:SC(1-G)=SC(1-G)+1:REM65
8080 FOR K=1 TO 100:NEXT K:REM25
8090 PRINT:PRINT:PRINTTAB(10)"SCORE DE";TAB(40)"SCORE DE":REM61
8100 PRINTTAB(10)N$(0);TAB(40)N$(1):REMDA
8110 PRINT:PRINTTAB(13)SC(0);TAB(43)SC(1):REM96
8120 FOR K=1 TO 100+RND(900):NEXT K:REM68
8130 RETURN:REM8D
9000 POKE AD,V:X=USR(C):REM41
9010 P=PEEK(14400):IF P<>8 AND P<>64 AND P<>72 THEN F=0 ELSE F=1
:REMC1
9020 RETURN:REM0F
9500 'FIN DE LA PARTIE '   BF
9510 CLS:REM31
9520 PRINT:PRINT:REM0D
9530 PRINT"LA PARTIE EST FINIE":REM02
9540 PRINT:PRINT:REMF9
9550 IF SC(0)=SC(1) THEN PRINT N$(0) " ET " N$(1) ",VOUS AVEZ FA
IT MATCH NUL":GOTO 9580:REMAA
9560 IF SC(0)>SC(1) THEN PRINTN$(0); ELSE PRINTN$(1);:REM97
9570 PRINT" VOUS AVEZ GAGNE .... BRAVO":REM6C
9580 PRINT:PRINT:REMD1
9590 PRINT"VOULEZ-VOUS FAIRE UNE AUTRE PARTIE ( O OU N )";:REMAE

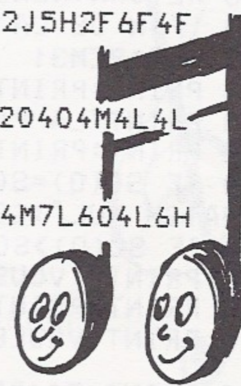
9600 INPUT R$:REM5C
```



```
9610 IF R$="N" THEN PRINT"AU REVOIR":END:REMFE
9620 IF R$="O" THEN SC(0)=0:SC(1)=0:GOTO 80:REM6E
9630 PRINT"REPONDEZ PAR O OU N S'IL VOUS PLAIT":GOTO 9590:REM48
10000 DATA 205,127,10,175,22,1,76,69,16,254,69,16,254,238,1,211,
255,13,32,243,21,32,239,201:REME1
11000 DATA"HYMNE A LA JOIE":REM9C
11010 DATA"L6L6M6O6O6M6L6J6H6H6J6L6L7J4J6L6L6M6O6O6M6L6J6H6H6J6L
6J7H4H6":REME3
11020 DATA"LE BOLERO":REMA9
11030 DATA"M6M4L4M4O4M4L4J4M6M4J4M6M4L4M4J4H4E4F4H8H4F4E4C4E4F4H
4J4H4H4":REM2D
11040 DATA"JESUS QUE MA JOIE DEMEURE":REMB8
11050 DATA"H4J4L4O4M4M4Q4O4O4T4S4T4O4L4H4J4L4M4O4Q4O4M4L4J4L4H4G
4H4J4C4E4J4M4L4J4L4":REMD8
11060 DATA"LA TRUITE":REM6B
11070 DATA"J4O4O4S4S4O4J5J2J6Q4O4N4L4J6J4O4O4S4S4O6J4O4N4L2N2O4I
4J6":REM99
11080 DATA"LETTRE A ELISE":REM3E
11090 DATA"Q4P4Q4P4Q4L4O4M4J6A4F4J4L6F4I4L4M6F4Q4P4Q4P4Q4L4O4M4J
6A4F4J4L6F4M4L4J6":REM34
11100 DATA"JEUX INTERDITS":REMBB
11110 DATA"Q6Q6Q6Q6O6M6M6L6J6J6M6Q6V6V6V6V6T6R6R6Q6O6O6Q6R6Q6R6Q
6U6R6Q6Q6O6M6M6L6J6L6L6L6L6M6L6J6J6J6J8":REMB3
11120 DATA"MENUET":REMCB
11130 DATA"H6M8L8M6O6H6J6F8E6F6C8E8A6H6M8L8M6O6H6J6F8E6F6C8F9":R
EM22
11140 DATA"TROMPETTES D'AIDA":REMF4
11150 DATA"D6I8K4D4K4M6M6M6M4N4I4M7K4I6K4M4M5K2I4K5M2M6K5M2M6I5K
2K8":REM81
11160 DATA"FRENCH CANCAN":REM17
11170 DATA"A4H4H4J4H4F4F4J4K4O4R4O4O4M4M6O4E4E4O4M4F4F4J4J4H4J4H
4J4H4J4H4J4H4J4H4H4F4F6":REMC3
11180 DATA"SE CANTO":REM50
11190 DATA"H6M6M6Q4Q4M6M6M4O4Q8Q6O8O4Q4R8R6Q6Q6O4M4O8O6M8":REM69

12000 DATA"AU CLAIR DE LA LUNE":REMDB
12010 DATA"H6H6H6J6L8J8H6L6J6J6H9H6H6H6J6L8J8H6L6J6J6H9J6J6J6J6E
8E8J6H6G6E6C6H6H6H6J6L8J8H6L6J6M6H9":REM2D
12020 DATA"J'AI DU BON TABAC":REM44
12030 DATA"C4E4G4C4E6E4G4H6H6G6G6C4E4G4C4E6E4G4H6J6C6O4Q4S4O4Q6Q
4S4T6T6S6S6O4Q4S4O4Q6Q4S4T6V6O6":REMEE
12040 DATA"SUR LE PONT D'AVIGNON":REMB4
12050 DATA"F4F4F6H4H4H6J4K4M4F4E4F4H4A4F4F4F6H4H4H6J4K4M4F4H4E4F
6F2F2F2F2H6F6F2F2F2F2H6F6":REM2E
12060 DATA"GENTIL COQUELICOT":REM2E
12070 DATA"J4J4H4F6J6M4M4H7J4J4H4F6J6M4M4H6H4J4K4M6M6K4J4H9H5F2H
4J4F4F4F4F4H5F2H4J4F4F4F6":REMC3
12080 DATA"TROIS JEUNES TAMBOURS":REM0D
12090 DATA"M6J5H2F7A2F5H2J5K2J6H4H6H5J2H7H2J5K2M5O2M5K2J5H2F6F4F
4F4A7A2F5H2J5K2J6H6F8":REMC9
12100 DATA"IL PLEUT BERGERE":REMA6
12110 DATA"L4O6L4O6L4H7C6H5G2H4J6J4L9L5J2L4M6M6O7L7O5Q2O4O4M4L4L
7J6":REMA3
12120 DATA"MALBROUGH":REM0C
12130 DATA"B4L6L4L6J4M7L4M4L4J4J4J4J4H4J4L7H6B4L6L4L6J4M7L6O4L6H
4J6G4H9":REMB8
15000 DATA "FIN":REM85
```







Des outils pour votre programmation (II)

Nous avons vu dans le précédent numéro, comment faire "avancer" la zone BASIC pour laisser une place susceptible de recevoir une routine en langage machine, par exemple. Cette zone privilégiée, présente l'avantage d'être auto-protégée tout en s'adaptant aux différentes configurations mémoire du TRS. Aujourd'hui, nous allons jeter un coup d'oeil sur une autre technique peu orthodoxe (qui fera surement bondir les programmeurs respectueux des traditions !) mais qui peut parfois être de quelque utilité : comment écrire des programmes BASIC en assembleur ..

L'exemple de programme dont la liste suit, a été réalisée avec l'éditeur/assembleur EDTASM. Il contient les lignes BASIC suivantes :

```
10 FOR N=1 TO 10
20 PRINT N ;
30 NEXT
```

Le début de la zone BASIC a été choisi pour s'adapter aux diverses configurations matérielles (7000H) et réalise ainsi, par la même occasion, un forçage de cette zone à une valeur différente de la valeur standard.

Nous avons vu précédemment (TRACE n° 2), que l'adresse du début de la zone BASIC était conservée dans le pointeur 40A4H/40A5H, alors que la fin était contenue en 40F9H/40FAH. Ces pointeurs seront chargés à l'initialisation du programme (lignes 420 à 450). Nous avons appris également, que la zone BASIC devait toujours être précédée d'un octet nul (ligne 60). Il suffit maintenant d'écrire le "corps" du programme BASIC sous forme de données (DEFB, DEFW et DEFM), en utilisant la représentation dite "compactée", telle qu'elle figure en mémoire. Dans celle-ci, chaque mot-clé du BASIC est représenté par une valeur prédéfinie appelée "token". Le mot clé PRINT, par exemple, est représenté par la valeur décimale 178. De cette technique, il découle un gain de place appréciable (un octet par mot-clé) et une rapidité dans l'exécution. C'est l'interprêteur BASIC qui, à la lecture des mots-clés (clavier, cassette ou disquette), effectue cette conversion.

Mais pour aller plus loin, il nous faut connaître la structure d'une ligne de programme en BASIC. La voici :

lien de chaîne (2 octets)
numéro de ligne (2 octets) en binaire
..... suite des mots-clés et des textes formant la ligne de programme (longueur variable)
octet nul signalant la fin de la ligne

- Le lien de chaîne contient l'adresse mémoire (poids faibles en tête) du début de la ligne suivante.
- Le numéro de la ligne BASIC, est codé en binaire (longueur fixe) sur deux octets (poids faibles en tête).
- Nous trouvons ensuite la ligne de programme proprement dite, dont la longueur dépend du texte et des mots-clés utilisés. Cette longueur est (en principe) limitée.
- La ligne se termine ensuite par un octet nul (valeur 0 en binaire).

La prochaine ligne, se trouvant immédiatement à la suite, aura une structure identique, et la fin physique du programme (ensemble de lignes) sera annoncée par un lien de chaîne nul (2 octets à zéro).

Cette fois-ci, nous en savons suffisamment pour écrire nos lignes BASIC en assembleur. Examinons par exemple, la ligne BASIC numéro 10 (lignes 100 à 180 du programme assembleur) et comparons sa structure avec celle du tableau ci-dessus.
Notez que le signe '=' possède aussi un "token". Il en est de même pour les signes '>' et '<'. Par contre, les noms de variables et les valeurs numériques sont codés en ASCII.

La fin du programme BASIC est codée sous la forme d'une valeur nulle de 16 bits (label FIN). L'adresse suivante (label LIBRE) signale le prochain emplacement libre de la mémoire, après la zone BASIC.

NOTA : Ce programme devrait tourner sur modèle 3, en prenant soin de remplacer l'instruction ' JP 6CCH' (ligne 460) par ' JP 1A19H'.

Consignes pratiques d'utilisation

Le programme étant assemblé et sauvé, deux cas se présentent : version cassette ou minidisquette.

1 - Version cassette :

. Charger le programme sous BASIC, par :
SYSTEM
*? INIT
(lecture de la bande)
et l'exécuter par :
*? /

Il suffit alors de frapper LIST pour retrouver le programme BASIC (et l'exécuter éventuellement).

2 - Version disquette :

. Charger le programme sous DOS par :
LOAD INIT/CMD
puis appeler BASIC et faire :
SYSTEM
*? / 28701

Cette dernière commande exécute un branchement au label INIT (adresse hexa 701DH) qui effectue la mise à jour des pointeurs de la zone BASIC. Dès cet instant, il suffit de faire LIST pour retrouver le programme sous une forme plus familière.

Alain Pinaud

```
                    00010 ;
                    00020 ; PRINCIPE D'ECRITURE D'UN PROGRAMME
                    00030 ;       BASIC... EN ASSEMBLEUR !
                    00040 ;
7000                00050         ORG     7000H           ; ORIGINE MEMOIRE
7000 00             00060         DEFB    0               ; OCTET-1 ZONE BASIC
                    00070 ;
                    00080 ; 10 FOR N=1 TO 10
                    00090 ;
7001 0D70           00100 LIGN10  DEFW    LIGN20          ; LIEN DE CHAINE
7003 0A00           00110         DEFW    10              ; NUMERO DE LIGNE
7005 81             00120         DEFB    129             ; CODE DU 'FOR'
7006 4E             00130         DEFM    'N'             ; N
7007 D5             00140         DEFB    213             ; =
7008 31             00150         DEFM    '1'             ; 1
7009 BD             00160         DEFB    189             ; TO
700A 31             00170         DEFM    '10'            ; 10
700C 00             00180         DEFB    0               ; FIN LIGNE 10
                    00190 ;
                    00200 ; 20 PRINT N ;
                    00210 ;
700D 1570           00220 LIGN20  DEFW    LIGN30          ; LIEN DE CHAINE
700F 1400           00230         DEFW    20              ; NUMERO DE LIGNE
7011 B2             00240         DEFB    178             ; PRINT
7012 4E             00250         DEFM    N;              ; N ;
7014 00             00260         DEFB    0               ; FIN LIGNE 20
                    00270 ;
                    00280 ; 30 NEXT
                    00290 ;
7015 1B70           00300 LIGN30  DEFW    FIN             ; LIEN DE CHAINE
7017 1E00           00310         DEFW    30              ; NUMERO DE LIGNE
7019 87             00320         DEFB    135             ; NEXT
701A 00             00330         DEFB    0               ; FIN LIGNE 30
                    00340 ;
                    00350 ; FIN DU PROGRAMME BASIC
                    00360 ;
701B 0000           00370 FIN     DEFW    0               ; DOUBLE ZERO
701D                00380 LIBRE   EQU     $               ; LA ZONE LIBRE...
                    00390 ;
                    00400 ; INITIALISATION DES POINTEURS BASIC
                    00410 ;
701D 210170         00420 INIT    LD      HL,LIGN10       ; ADRESSE LIGNE 10
7020 22A440         00430         LD      (40A4H),HL      ; --> POINTEUR DEBUT
7023 211D70         00440         LD      HL,LIBRE        ; ADRESSE LIBRE
7026 22F940         00450         LD      (40F9H),HL      ; --> POINTEUR FIN
7029 C3CC06         00460         JP      6CCH            ; RETOUR AU BASIC
701D                00470         END     INIT
00000 TOTAL ERRORS
33726   TEXT AREA BYTES LEFT


FIN     701B 00370    00300
INIT    701D 00420    00470
LIBRE   701D 00380    00440
LIGN10  7001 00100    00420
LIGN20  700D 00220    00100
LIGN30  7015 00300    00220
```

L'ordinateur à la maternelle

L'ordinateur est un peu la "lampe d'Aladin" de notre époque : selon notre humeur (mais aussi selon la cassette avec laquelle nous le frottons !), il saura se transformer en comptable, en comparaison de jeu ou en patient pédagogue.
Ce modèle 1, par exemple, équipé d'une carte 80-GRAFIX (générateur de caractères programmable), s'est discrètement introduit dans une école maternelle. Mais voici l'institutrice, profitons-en pour lui poser quelques questions ...

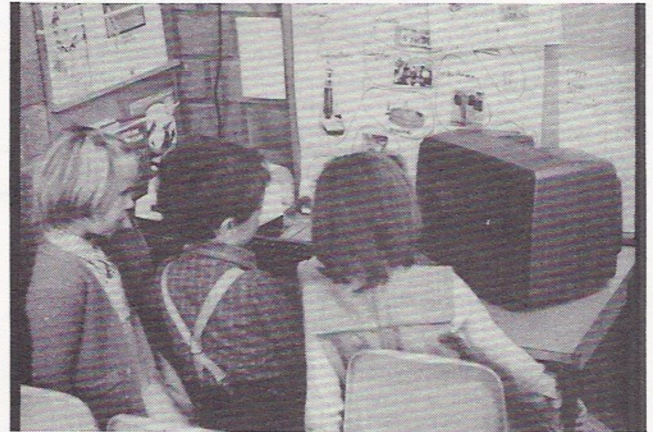
- Quelles raisons vous ont amené à utiliser un ordinateur dans une classe d'école maternelle ?

- Mon mari (qui est professeur de mathématiques, cela à son importance) ayant acheté un TRS-80, a commencé par écrire des programmes mathématiques, puis quelques jeux pour enfants : il s'agissait de trouver le nombre d'animaux apparaissant sur l'écran ou le nombre de trous fait par une chenille dans une feuille ... L'enthousiasme de nos enfants, grimpés sur le bureau pour compter les animaux ou embrasser la chenille, m'a conduit à penser qu'il serait dommage de se priver, en classe, d'un outil aussi stimulant. Je dois ajouter que cela a aussi été, pour moi, un moyen d'apprivoiser "Thérèse", devenue ma plus grande rivale aux yeux de mon mari !

- Hum ... Trace se préoccupera plus tard de ces questions ... Mais auparavant, pouvez-vous nous dire quels genres de programmes vous utilisez en classe ?

- Jusqu'à présent, nous en avons utilisé quatre, intitulés : "Labyrinthe", "Autant", "Plus" et "Construction".
Comme son nom l'indique, le premier programme créé des labyrinthes. Ils sont de dimensions variables, mais n'ont qu'une seule issue. L'enfant s'y déplace grâce aux touches fléchées, et le chemin qu'il parcourt est visualisé ou effacé s'il revient sur ses pas. Il est impossible de traverser les murs du labyrinthe. Une fois la sortie atteinte, une petite musique récompense le succès puis le labyrinthe et le chemin parcouru sont dessinés sur imprimante.
Dans le programme "Autant", deux collections d'objets sont dessinées, l'une à droite et l'autre à gauche de l'écran. L'enfant fait apparaître un lien entre objets de droite et objets de gauche, pour mettre en évidence la relation entre les deux collections. Puis il répond (par oui ou par non) à la question lui demandant s'il y a autant d'objets à droite qu'à gauche.

Le programme "Plus" reprend le même principe, mais plusieurs questions sont posées : combien y a t-il d'objets à droite, à gauche, de quel côté y en a t-il le plus, combien y en a t-il en plus ...
Enfin, dans le programme "Construction", hui pièces de formes différentes sont à la disposition des enfants (un disque, deux rectangles, un carré, quatre triangles d'orientations différentes). Chaque pièce a un numéro grâce auquel l'enfant l'appelle. Il peut alors déplacer la pièce choisie grâce aux touches fléchées, et construire ce qu'il veut (dans la pratique, il y a beaucoup de bonshommes et de maisons). Lorsqu'il est terminé, le dessin réalisé à l'écran est tracé sur imprimante. Bien entendu, l'enfant dispose de possibilités d'effacement et peut appeler une pièce autant de fois qu'il le désire.



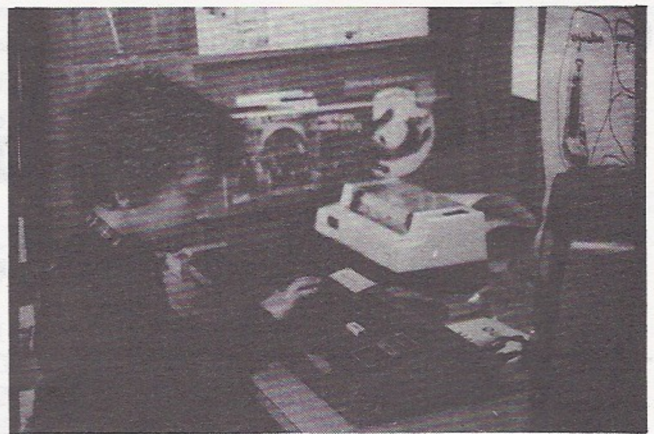
- Comment fonctionnez-vous : individuellement ou en groupe ? L'ordinateur est-il en permanence dans votre classe ?

- Au départ, je n'ai disposé de l'ordinateur qu'une demi-journée par semaine. Mon mari a pris des groupes de deux à cinq enfants pour les initier au maniement de l'appareil (touche blanche indispensable pour la réponse, touche flèche à gauche pour effacer ...) et leur expliquer comment utiliser un programme (tout utilisateur introduit d'abord son nom ...).
Maintenant, le maniement de l'outil étant assimilé, l'ordinateur est utilisé une journée par semaine, soit par groupe de trois (pour "Plus" et "Autant"), soit individuellement ("Labyrinthe", "Construction"). Parfois, les enfants décident eux-mêmes du mode d'utilisation. Ainsi un jour, deux enfants se sont répartis les touches fléchées pour faire ensemble le labyrinthe ... Puis le jeu s'est fait à quatre, chacun sa touche : bon travail d'équipe spontané ... jusqu'au moment où la règle a été transformée en "tout faire pour bloquer les autres" ... !

- Pouvez-vous dresser un bilan provisoire de votre expérience ?

- J'ai utilisé l'ordinateur dans deux domaines assez différents. Le premier domaine, c'est celui du contrôle de l'acquisition des connaissances (programmes "Plus" et "Autant"). Là, les notions mises en jeu (autant, nombre ...) ont déjà fait l'objet de nombreuses activités en classe, et l'ordinateur ne peut pas remplacer ces activités indispensables. Par contre, au niveau pédagogique, l'évaluation immédiate de la réponse et le correction éventuelle sont d'un grand intérêt. De plus, ce type de travail favorise l'autonomie de l'enfant par rapport à l'adulte. Dans la mesure du possible, questions et

réponses sortent sur imprimante : cela me permet de contrôler le niveau d'acquisition.
Second domaine d'utilisation,: il ne s'agit plus de contrôle des connaissances, mais plutôt d'un jeu où l'enfant doit avoir un rôle particulièrement actif ("Labyrinthe", "Construction"). Ces programmes concernent les notions d'orientation dans l'espace (droite - gauche, haut - bas, orientation des triangles dans le jeu de construction). Ces notions font aussi l'objet d'activités en classe, mais là, l'ordinateur apporte une autre présentation, ce qui peut permettre une approche différente.
A l'usage, les programmes ont un intérêt que je n'imaginais pas au départ. Ainsi, dans les labyrinthes, comme il n'y a qu'une seule issue possible, un enfant ne peut que réussir. Et toute expérience positive est réinvestie par l'enfant dans un autre domaine.
Autre intérêt non prévu au départ. L'ordinateur est un intermédiaire entre la manipulation et la représentation. Ce n'est peut-être pas clair : je précise. Le programme "Construction", ressemble à un puzzle. Dans un puzzle, l'enfant manipule des pièces pour représneter quelque chose. Avec le programme, l'enfant manipule les pièces de façon particulière (droite, gauche, haut, bas) et détermine lui-même ce qu'il veut représenter ... mais ce n'est pas non plus un dessin. C'est un intermédiaire qui favorise l'anticipation : il faut d'abord se représenter ce qu'on veut construire pour ensuite appeler les pièces utiles.
Encore quelques remarques au niveau pédagogique. Tout d'abord, l'ordinateur autorise le droit à l'erreur. Dans la plupart des exercices faits en classe, l'erreur commise subsiste sur le cahier, sur la feuille polycopiée ... Avec l'ordinateur, l'erreur commise laisse beaucoup moins de trace.
Dernière remarque : l'ordinateur est un outil très motivant (et pas seulement parce que Sandrine en a commandé un au Père Noël !). Des enfants particulièrement instables se sont révélés capables d'une attention soutenue et assez longue pour sortir d'un labyrinthe. Autre aspect motivant et important pour une section de grands : l'ordinateur motive l'intérêt de l'enfant pour l'expression écrite. Avant toute utilisation, il faut taper son nom, décoder la question posée ...



- Est-ce à dire que l'ordinateur est un outil indispensable ?
- A l'heure actuelle, sûrement pas ! Il ne peut pas fournir certaines expériences et certaines manipulations indispensables aux enfants. Toutefois, son aspect motivant et l'originalité des services qu'il peut rendre (contrôle, droit à l'erreur, développement de la créativité, possibilités illimitées ...) en font un allié très positif de l'institutrice.

Propos recueillis par Cyril Grandpierre.

bricolage



Minidisquettes double faces sur Modèle 1

On trouve actuellement dans le commerce, des unités de minidisquettes à double-face, c'est-à-dire munies de deux têtes de lecture/écriture, permettant d'exploiter simultanément les deux faces des disquettes 13 centimètres sans retournement de la pochette.

Physiquement, ces unités occupent l'emplacement d'une unité simple-face, tout en ayant une capacité double. En utilisant un dispositif doubleur de densité, on atteint 360 K octets par unité, soit l'équivalent de quatre unités simple-face. Il est bon d'y réfléchir ! Quant à leur prix, il est de loin inférieur à deux unités (moins de 3000 francs sans carrosserie et sans alimentation). Néanmoins, il faut savoir qu'une unité double-face n'est pas toujours égale à deux unités simple-face, il suffit de faire un "backup" (recopie de sauvegarde) pour s'en rendre compte !

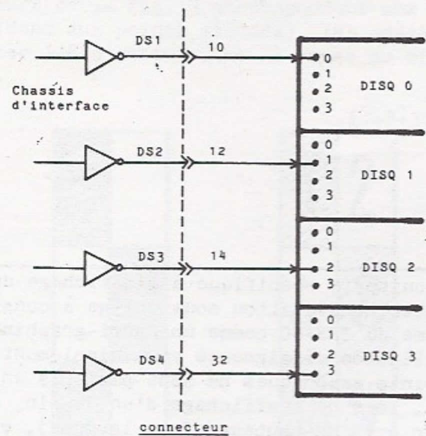
Il y a deux manières d'exploiter les unités double-face :

a) Le circuit intégré qui contrôle les disquettes (FDC) possède un signal permettant de distinguer l'une des deux faces (SIDE SELECT) et il est donc possible au logiciel qui pilote ce circuit (driver ou "programme pilote") de sélectionner la face choisie.

b) Le FDC et le pilote ne sont pas prévus pour gérer les unités double-face. Celles-ci sont donc considérées par le logiciel comme étant chacune deux unités simple face.

Ce dernier cas est celui du modèle 1.
En version standard, celui-ci peut piloter quatre unités simple-face. Avec la petite modification exposée ci-dessous, il lui sera possible de contrôler deux unités double-face : la première unité correspondant aux adresses :0 (face A) et :1 (face B), la seconde aux adresses :2 (face A) et :3 (face B).

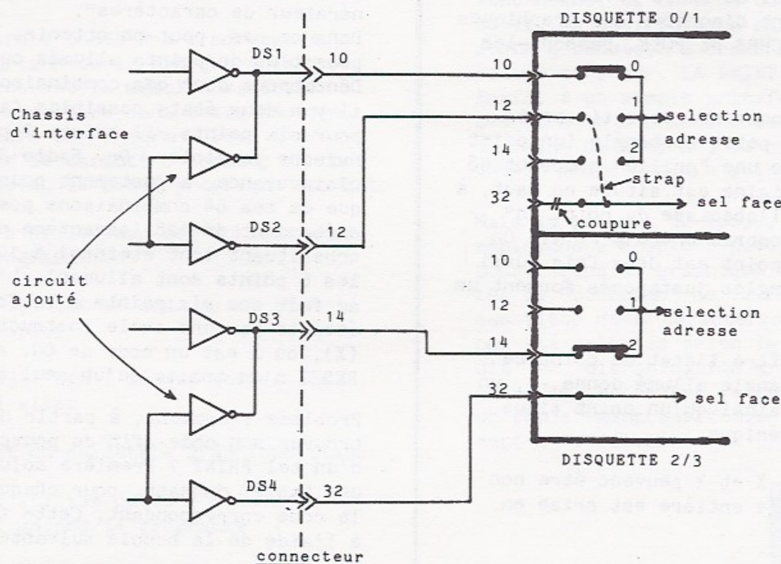
Schéma de principe en standard (4 unités simple-face)

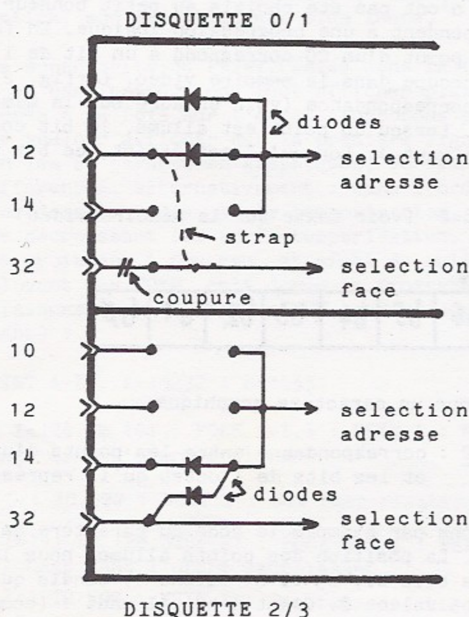


Avec la modification que nous vous proposons, les signaux DS1 et DS2 doivent sélectionner l'unité 0 (head load), DS2 jouant le rôle "SIDE SELECT" ; DS3 et DS4 doivent sélectionner l'unité 1 (head load), DS4 jouant le rôle "SIDE SELECT". Les combinaisons logiques sont donc les suivantes :

Sélection	Adresse	Unité double-face
DS1	:0	Unité 0 face A
DS2	:1	Unité 0 face B (unité 1 en standard)
DS3	:2	Unité 1 face A (unité 2 en standard)
DS4	:3	Unité 1 face B (unité 3 en standard)

Schéma de principe après modification





Remarquez la coupure de piste à effectuer sur la première unité (SIDE SELECT). Si vous préférez ne pas "charcuter" le circuit imprimé, vous pouvez toujours déposer du vernis ou coller un petit morceau de ruban adhésif sur le contact numéro 32 du connecteur.
Le circuit intégré utilisé est du type 7438 ou 7416. Dans notre ancien modèle d'interface d'extension, il était possible d'employer le circuit libre Z39 (entrées 4, 5, sortie 6 et entrées 12, 13, sortie 11). Qu'en est-il sur les nouveaux modèles ? Aussi nous vous conseillons d'utiliser un circuit extérieur.
Pour ceux qui ne désirent pas modifier leur châssis d'extension, il existe une autre solution qui consiste à utiliser des diodes au germanium (OA 85 par exemple), à la place des straps de sélection d'adresses des disquettes (voir figure ci-dessous). Dans ce cas, les modifications précédentes ne doivent pas être appliquées. Les diodes seront soudées sur des petits contacts venant s'enficher à la place des straps de sélection d'adresse.
Dernier détail : le câble de liaison interface/disquettes livré par Tandy ne peut pas être utilisé tel quel, certains signaux de sélection étant absents (contacts retirés dans les connecteurs).

Jean-Pierre du Tillet

Les graphiques sur TRS

Les graphiques ... Mot magique, argument publicitaire, cette caractéristique technique est souvent mise en exergue dans la description d'un OI. Apparemment, ceux du vieux TRS-80 modèle 1 sont bien pauvres, face à certains nouveaux venus dont les publicités proclament fièrement : "graphiques couleurs en haute résolution" (et en lettres microscopiques : "moniteur couleur, interface SECAM ou RVB en option"). Et pourtant, nous allons voir qu'avec quelques astuces, on parvient à obtenir des affichages rapides et spectaculaires...

Selon notre Bible, le Manuel du BASIC Level II, les cinq instructions permettant d'accéder aux graphiques sont : SET, RESET, POINT, CHR$ et POKE. Passons-les en revue.

SET(X,Y) et RESET(X,Y) permettent respectivement d'allumer et d'éteindre un petit rectangle (un point graphique élémentaire) dans une "grille" comptant 48 lignes de 128 points. L'origine est située en haut, à gauche de l'écran ; X est l'abscisse du point (n° de colonne) et doit être compris entre Ø et 47. On peut constater que chaque point est deux fois aussi haut que large (deux rectangles justaposés forment un carré).

POINT(X,Y) permet de connaître l'état du point de coordonnées (X,Y). Un rectangle allumé donne -1, un rectangle éteint donne 0, ainsi qu'un point situé dans un caractère alphanumérique.

A noter que les valeurs de X et Y peuvent être non entières : seule leur partie entière est prise en compte.

CHR$(X) fournit un caractère dont le code est X. Le code peut être emprunté au standard ASCII. Mais ce standard et celui employé par le TRS ne coïncident que jusqu'à X=127. Au-delà, on tombe dans le domaine des caractères graphiques (CG) propres au TRS, qui sont des blocs de 6 points élémentaires, disposés en 2 x 3 (fig. 1).

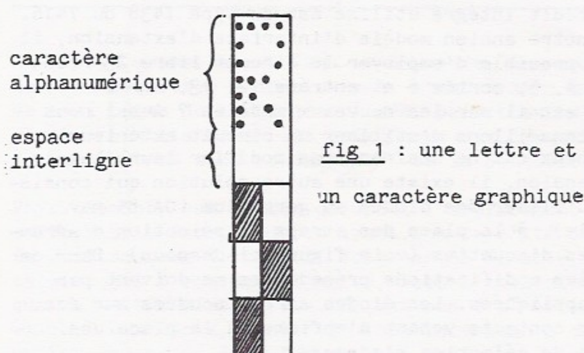


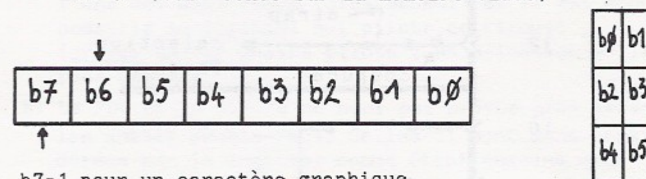
fig 1 : une lettre et un caractère graphique

Arrivés à ce point, faisons une petite incursion dans les circuits de l'interface vidéo du TRS-80. Celle-ci ne sait afficher que des caractères, n'ayant pas de circuiterie spécifique à l'affichage de points. Cette constatation nous oblige à considérer le graphisme du TRS-80 comme un "semi-graphisme". Cette appellation dédaigneuse veut simplement dire que les points graphiques ne sont pas pris en compte séparément, lors de l'affichage d'un dessin, mais groupés par 6 (3 en hauteur, 2 en largeur), et que ce groupe est considéré par l'interface comme un caractère : sa forme est reconstituée par une circuiterie spéciale (codeur graphique) de l'interface vidéo, contrairement aux caractères alphanumériques dont le dessin est contenu dans une ROM spéciale dite "générateur de caractères".
Dans ce cas, peut-on obtenir, toutes les combinaisons possibles de points allumés ou éteints ? Mais oui. Dénombrons donc ces combinaisons : pour chaque point, il y a deux états possibles (allumé ou éteint) ; donc, pour six points, il existe 2 puissances 6= 64 combinaisons possibles. Or, Radio Shack, dans son infinie clairvoyance, a justement pourvu ledit codeur graphique de ces 64 combinaisons possibles, dont les codes s'étendent de 128 (caractère noir, les 6 points le constituant sont éteints) à 191 (caractère blanc, les 6 points sont allumés). L'intérêt des CG tient au fait que six points à la fois sont allumés ou éteints par une seule instruction du type PRINT CHR$ (X), où X est un code de CG, alors qu'un SET ou un RESET n'en traite qu'un seul à la fois.

Problème : comment, à partir de la forme d'un CG, trouver son code afin de pouvoir l'imprimer à l'aide d'un tel PRINT ? Première solution : se référer à une table, donnant, pour chaque caractère graphique, le code correspondant. Cette table peut être obtenue à l'aide de la boucle suivante :

```
FOR I=128 TO 191 : PRINT I;":"; CHR$(I) ;: NEXT I
```

Deuxième solution (qui est peut-être la plus rapide avec un peu d'habitude) : on remarque que les codes des CG n'ont pas été choisis au petit bonheur, mais correspondent à une progression logique. En fait, chaque point d'un CG correspond à un bit de l'octet qu'il occupe dans la mémoire vidéo. La fig. 2 donne cette correspondance (voir encadré sur la mémoire vidéo). Lorsqu'un point est allumé, le bit correspondant vaut 1. Lorsqu'il est éteint, ce bit vaut Ø.



b7=1 pour un caractère graphique

fig 2 : correspondance entre les points d'un CG et les bits de l'octet qu'il représente

Calculons par exemple le code du caractère de la fig. 1. La position des points allumés nous indique que les bits bØ, b3 et b4 valent 1, tandis que b1, b2 et b5 valent Ø. Quant à b7, il vaut 1 (comme pour tous les CG), et b6 vaut Ø. Le code s'écrit donc

1001 1001 en binaire, soit 153 dans notre bon vieux système décimal. Cela revient finalement à relever les valeurs de la fig. 3 correspondant aux bits valant 1 (donc aux points allumés), les additionner, et ajouter 128 à cette somme (à cause du bit b7 valant 1).

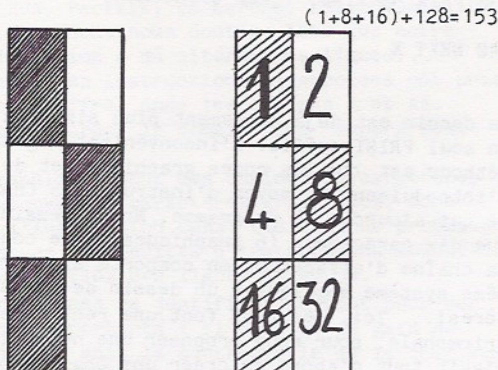


fig 3

le CG de code 153... ... et la façon de calculer ce code

POKE A,X permet d'écrire l'octet X à l'adresse A, dans la RAM. Et une portion de la RAM qui nous intéresse particulièrement est la mémoire vidéo. En écrivant le code d'un caractère dans cette mémoire, nous le voyons apparaître sur l'écran. Pour l'effacer, on le remplace par le caractère espace, de code 32. Essayons, par exemple, le petit programme suivant:

```
10 DEFINT A-Z : Rem travailler sur des entiers accélère l'exécution du programme

20 B=191 : E=32 : Rem 191 est le code du CG blanc, et 32 est celui de l'espace.

30 FOR I=15360 TO 16383 : Rem I décrit toutes les adresses de la mémoire vidéo

40 POKE I,B : Rem écrit un caractère graphique blanc

50 FOR J=1 TO 5 : NEXT J : Rem cette boucle de temporisation ralentit un peu le programme

60 NEXT I
```

Un POKE prend en charge six points à la fois. Il est jusqu'à deux fois plus rapide qu'un PRINT équivalent, car, vu la versatilité de cette dernière instruction, son exécution est plus lente.

Mais la supériorité du PRINT réside dans le fait qu'une de ces instructions peut afficher plusieurs caractères à la fois sur l'écran. Supposons par exemple vouloir afficher sur la première ligne de l'écran les 64 caractères graphiques, de deux manières différentes, alternativement : dans l'ordre des codes croissants, puis, après temporisation, dans l'ordre décroissant (et après temporisation, dans l'ordre croissant à nouveau, et ainsi de suite). En employant les POKE, cela donne le programme suivant (faisons tout d'abord un NEW pour effacer le précédent) :

```
10 DEFINT A-Z : A=15232 : B=15551

20 FOR I=128 TO 191 : POKE A+I,I : NEXT I : Rem affichage dans l'ordre croissant

30 FOR I=1 TO 500 : NEXT I : Rem temporisation

40 FOR I=191 TO 128 STEP -1 : POKE B-I,I : NEXT I : Rem affichage dans l'ordre décroissant

50 FOR I=1 TO 500 : NEXT I : Rem temporisation

60 GOTO 20 : Rem etc..., jusqu'à un BREAK
```

Voyons à présent comment employer les PRINT, avec le programme suivant (faisons un NEW avant de taper cette nouvelle version) :

```
10 CLEAR 200 : DEFSTR A-B : DEFINT C-Z : Rem le CLEAR réserve de la mémoire pour les chaînes A et B

20 FOR I=128 TO 191 : A=A+CHR$(I) : NEXT I : Rem A est la chaîne des codes croissants

30 FOR I=191 TO 128 STEP -1 : B=B+CHR$(I) : NEXT I : Rem B est la chaîne des codes décroissants

40 PRINT @ 0,A : : Rem A est affichée

50 FOR I=1 TO 500 : NEXT I : Rem temporisation

60 PRINT@ 0, B ; : Rem B est affichée

70 FOR I=1 TO 500 : NEXT I : Rem temporisation

80 GOTO 40 : Rem etc...
```

Cette fois, l'affichage de la ligne de CG est presque instantané, et en tous cas trop rapide pour être suivi. La modification par rapport à la version précédente a consisté à créer les deux lignes de CG avant de les afficher. Le PRINT se résume alors, pour le Basic, à un simple transfert de CG depuis les adresses où sont stockées A et B jusqu'aux adresses de la 1e ligne de l'écran, ce qui ne représente que quelques microsecondes à peine...

Mais la plupart du temps, les graphiques que nous employons s'écartent plutôt de ce style abstrait. Considérons un graphique utilisable dans un jeu, tel le "bonhomme" de la fig. 4. Notre bonhomme est constitué de 6 caractères graphiques, dont les codes ont été calculés selon la méthode de la fig. 3. Qui dit jeu, dit animation graphique, aussi rapide que possible. Notre but va être d'afficher ce bonhomme à un emplacement quelconque de l'écran, aussi rapidement et facilement que possible.

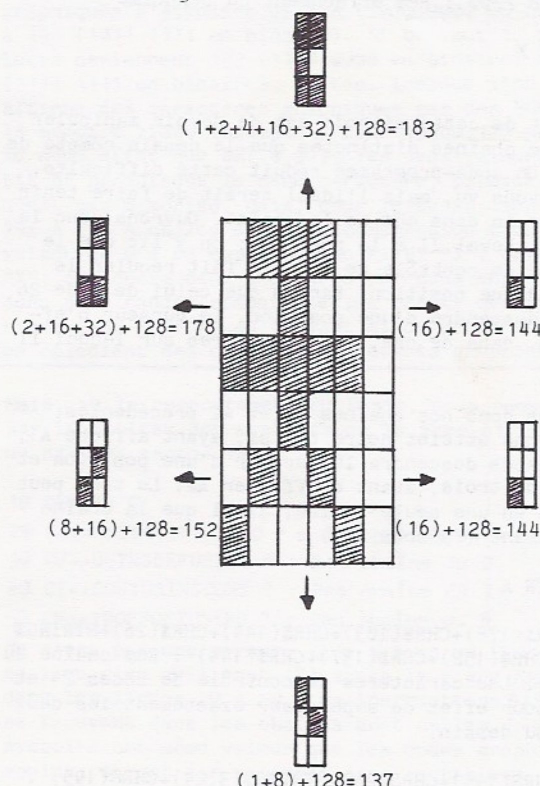


fig 4 : un "bonhomme en six caractères

Une méthode simple consiste à créer deux chaînes de CG, les moitiés hautes et basses du bonhomme (faisons un NEW avant de taper cette nouvelle version) :

10 DEFSTR A : Rem pour plus de commodité, mieux vaut définir A comme une chaîne

20 A1=CHR$(178)+CHR$(183)+CHR$(144) : Rem moitié haute

30 A2=CHR$(152)+CHR$(137)+CHR$(144) : Rem moitié basse

Puis, afin de déplacer notre bonhomme sur l'écran, écrivons deux sous-programmes, l'un qui affichera le bonhomme, l'autre qui l'effacera (car lorsqu'il aura été affiché à une nouvelle position, il faudra l'effacer de la précédente). Il faut donc d'abord créer une chaîne d'effacement :

40 A3=CHR$(195) : Rem ce caractère est affiché comme trois espaces (code de compression), mais n'occupe qu'un seul octet

Les deux sous-programmes seront :

999 END : Rem il faut isoler les routines du corps
1000 PRINT@ X,A1 ;: PRINT @ X+64,A2 : Rem A2 est affichée à 64 positions de A1, donc juste au-dessous

1010 RETURN

2000 PRINT@ X, A3 ; : PRINT@ X+64, A3 : Rem : on efface ainsi les moitiés hautes et basses

2010 RETURN

Utilisons maintenant ces sous-programmes, en baladant notre bonhomme sur l'écran :

100 FOR X=0 TO 800

110 GOSUB 1000 : Rem affichage du bonhomme

120 FOR I=1 TO 10 : NEXT I : Rem temporisation

130 GOSUB 2000 : Rem effacement du bonhomme

140 NEXT X

Le défaut de cette méthode est de devoir manipuler autant de chaînes distinctes que le dessin compte de lignes. Un sous-programme réduit cette difficulté, nous l'avons vu, mais l'idéal serait de faire tenir notre dessin dans une seule chaîne. Ouvrons donc le Manuel du Level II à la page C1 : on y lit que le caractère de contrôle de code 24 fait reculer le curseur d'une position, tandis que celui de code 26 le fait descendre d'une position. Le curseur n'efface pas, dans ce cas, les caractères sur lequel il passe.

Reprenons donc nos chaînes A1 et A2 précédentes. Nous aurons atteint notre but si, ayant affiché A1, nous faisons descendre le curseur d'une position et reculer de trois, avant d'afficher A2. Le tout peut s'écrire en une seule chaîne, ainsi que la chaîne d'effacement (faisons NEW) :

10 DEFSTR A

20 A=CHR$(178)+CHR$(183)+CHR$(144)+CHR$(26)+STRING$(3,24)+CHR$(152)+CHR$(137)+CHR$(144) : Rem chaîne du bonhomme. Les caractères de contrôle de codes 24 et 26 ont pour effet de superposer exactement les deux lignes du dessin.

30 AE=CHRS(195)+CHRS(26)+STRINGS(3,24)+CHRS(195) : Rem chaîne d'effacement

Affichons les chaînes :

100 FOR X=1 TO 800

110 PRINT@ X,A; : Rem affichage du bonhomme

120 FOR I=1 to 10 : NEXT I : Rem temporisation

130 PRINT@ X,AE; : Rem effacement

140 NEXT X

Le dessin est déjà nettement plus simple à manipuler: un seul PRINT suffit. L'inconvénient majeur de cette méthode est que les codes graphiques et de contrôles s'introduisent au moyen d'instructions CHR$, ce qui alourdit le programme. Notre dessin ne compte que dix caractères (6 graphiques, 4 de contrôle), et la chaîne d'effacement en comporte six. Imaginez le même système appliqué à un dessin de cinquante caractères!... Ici, les POKE font une rentrée en scène triomphale, pour nous proposer une nouvelle ruse. Il s'agit tout d'abord de créer une chaîne contenant n'importe quoi, pourvu qu'elle comporte autant de caractères que le dessin recherché. Notre dessin comporte dix caractères, et la chaîne d'effacement en comporte six. Les chaînes seront donc (faisons un NEW auparavant) :

10 DEFSTR A

20 A="0123456789" : Rem dix caractères quelconques, ici faciles à compter

30 AE="012345" : Rem six caractères quelconques

A ce stade, repérons les emplacements E1 et E2 où sont mémorisées les chaînes A et AE (voir Manuel page 8/9) :

40 V=VARPTR(A) : E1 = PEEK(V+1) + 256*PEEK(V+2)

50 V=VARPTR(AE) : E2=PEEK(V+1) + 256*PEEK(V+2)

Après exécution de ces lignes, E1 et E2 sont les adresses respectives du 1e caractère de A et AE. Nous allons donc pouvoir remplacer le contenu sans intérêt de ces chaînes par notre dessin et notre chaîne d'effacement, en "pokant" à ces emplacements les codes voulus, lesquels peuvent être contenus dans des lignes de DATA :

60 RESTORE : Rem initialise le pointeur de DATA

70 FOR I=0 TO 9 : Rem A est longue de 10 caractères

80 READ J : POKE E1+I,J : Rem les valeurs lues dans les DATA de la ligne 100 sont pokées dans A

90 NEXT I

100 DATA 178,183,144,26;24,24,152,137,144 : Rem nous reconnaissons les valeurs des codes graphiques et de contrôle du bonhomme

110 FOR I=0 TO 5 : Rem AE est longue de 6 caractères

120 READ J : POKE E2+I,J : Rem les valeurs lues en 140 sont pokées dans AE

130 NEXT I

140 DATA 195,26,24,24,24,195 : Rem valeurs des codes de la chaîne d'effacement

Utilisons à présent ces chaînes, comme précédemment :

150 FOR X=0 TO 800

160 PRINT@ X,A ; : Rem affichage dessin

```
170 FOR I=1 TO 10 : NEXT I : Rem temporisation
180 PRINT@ X,AE : : Rem effacement dessin
190 NEXT X
```

Faisons un RUN. Parfait, ça marche. Faisons ensuite LIST 20-30, car nous nous doutons bien que notre petite manipulation a dû altérer ces lignes... Effectivement, des instructions bizarroïdes ont pris la place des chiffres dans les chaînes A et AE. Nous venons d'aborder le domaine des "supergraphiques"...
Les supergraphiques sont des chaînes de caractères où l'on a stocké des caractères graphiques, sans passer par l'instruction CHR$. Ces chaînes produisent lors du listage, des suites d'instructions incohérentes, comme celles des lignes 20 et 30.
Les supergraphiques se justifient surtout pour des dessins assez importants : il faut qu'un gain de place en mémoire soit la contrepartie de la complication qu'ils introduisent. Pour introduire les codes graphiques et de contrôle dans les chaînes, on emploie surtout la méthode du programme précédent : créer des chaînes inutiles, mais de même longueur que les graphiques voulus, repérer leurs emplacements à l'aide de VARPTR, et, au moyen de boucles, on "poke" dans les chaînes les codes graphiques contenus dans des DATA.

Mais pourquoi obtenons-nous, en listant ces chaînes, des instructions pêle-mêle ? Cela tient à la façon dont le Level II mémorise les instructions d'un programme. Lorsqu'on tape "Enter" à la fin d'une ligne de programme, le Level II ne la mémorise pas telle qu'elle a été frappée (sous forme de caractères ASCII). Il commence par la parcourir, à la recherche d'instructions BASIC reconnaissables. Dans cette phase, il ne se préoccupe nullement de la syntaxe. Mais lorsqu'il reconnait un mot-clé de Basic, il lui substitue aussitôt son "token" (mot anglais que je me risquerais à traduire par "symbole" plutôt que "jeton"). Le symbole d'une instruction BASIC est un octet que le Level II mémorise à la place de l'instruction, pour économiser de la place en mémoire. Lors du listage du programme, l'instruction réapparait sous sa forme lisible (suite de caractères ASCII). Ainsi, par exemple, PRINT est remplacé par l'octet 178. Et inversement, lors du listage, tout octet valant 178 sera remplacé par le mot PRINT. Seules exceptions : tout ce qui se trouve entre guillemets ou derrière un REM est laissé tel quel. Et lorsque, malicieusement, nous mettons des codes graphiques dans des chaînes où il ne devrait pas y en avoir normalement, le BASIC les liste comme des instructions. Les symboles d'instructions s'étendent de 128 à 255, et les codes des caractères graphiques de 129 à 191.

On peut mettre à profit cette propriété pour les graphiques n'utilisant pas de caractères de contrôle, par exemple pour les dessins tenant sur une seule ligne. Supposons par exemple vouloir écrire un sous-programme posant à l'utilisateur la question "oui ou non", et revenant au programme principal dès que l'une des touches O ou N a été enfoncée. Pour rendre le programme plus attrayant, faisons en sorte d'afficher ladite question en gros caractères. La fig. 5 donne un graphisme possible pour les lettres O, U, I et N que comporte la phrase "oui ou non", ainsi que les codes des CG qui en résultent.

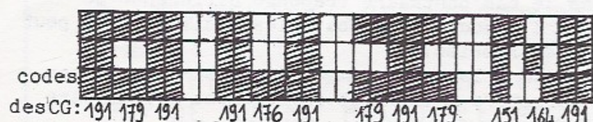


fig 5 : graphisme des lettres O, U, I et N. Chaque lettre occupe 3 CG, plus un espace de séparation

On pourrait définir quatre chaînes, une pour chaque lettre, au moyen de CHR$:

```
10 DEFSTR C : Rem C. comme chaîne. Pourquoi
toujours A ?
20 CO=CHR$(191)+CHR$(179)+CHR$(191) : Rem chaîne du
O
30 CU=CHR$(191)+CHR$(176)+CHR$(191) : Rem chaîne du
U
 Etc...
```

LA MEMOIRE VIDEO

On appelle ainsi le segment de mémoire occupant les adresses 15360 à 16383. Chaque caractère de l'écran (il y en a 1024, répartis en 16 lignes de 64 caractères) occupe un octet de la mémoire vidéo. Ainsi, l'adresse 15360 correspond au caractère de position 0 (en haut, à gauche); et l'adresse 16383 correspond au caractère de position 1023 (en bas, à droite). Hum, un octet, avons-nous dit ? Pas tout à fait : si vous ouvrez votre TRS-80, vous dénombrerez 7 boîtiers de 1024 bits pour la mémoire vidéo (et ferez sauter la garantie par la même occasion). Donc, chaque caractère occupe en réalité sept bits (un septet?) de la mémoire vidéo, mais on le compte toujours comme un octet. Simplement, le 7e bit de l'octet n'est pas mémorisé. Le caractère est alphanumérique si le bit b7 (le 8e) vaut 0, et graphique si ce bit vaut 1 : ainsi les circuits de l'interface vidéo font-ils la distinction.

Afficher un caractère se fait donc en mettant dans la mémoire vidéo le code de ce caractère, à la position où on veut le voir. L'instruction POKE est un moyen évident d'écrire un octet en mémoire, et PRINT en est un plus commode. Si N est le code d'un caractère, compris entre 0 et 191, et si X est un entier compris entre 0 et 1023, les instructions PRINT@ X,CHR$(N) et POKE 15360+X,N ont exactement le même effet.

Ce fameux bit b6 n'est pas mémorisé, avons-nous dit. Mais lorsqu'on opère sur des caractères graphiques, mieux vaut le considérer comme valant zéro. En effet, lorsque b6=0, les valeurs des codes des caractères graphiques s'étendent de 128 (1000 0000 en binaire) à 191 (1011 1111 en binaire). Si b6 vaut 1, ces valeurs deviennent 192 (1100 0000 en binaire) à 255 (1111 1111 en binaire). Certes, lorsque l'on affiche des caractères graphiques par des POKE dans la mémoire vidéo (exemple : POKE 15360,129 produit le même effet que SET(0,0)), les deux valeurs du bit b6 produisent le même caractère, puisque ce bit n'est pas mémorisé. Mais, pour le Basic, les codes 192 à 255 sont les codes de compression d'espace, valant de 0 à 63 espaces : un PRINT "X"; CHRS(194); "X"; affichera deux X séparés par deux espaces. C'est donc pour rendre compatible les effets des instructions CHRS et POKE qu'il vaut mieux prendre b6=0, en calculant des codes de caractères graphiques.

Mais, vu le comportement du Basic, nous pouvons utiliser le tableau des symboles de la page 41. Faisons un NEW, et tapons :

```
10 DEFSTR C
20 CO=+USINGCONTUSING " : Rem chaîne du O
30 CU=+USINGDEFUSING " : Rem chaîne du U
40 CI=+CONTUSINGCONT " : Rem chaîne du I
50 CN=+TROFFGETUSING " : Rem chaîne du N
```

Ce n'est pas une erreur, il y a bien un Plus (+) au lieu d'un guillemet (") après le signe Egal (=) dans les lignes 20 à 50. Les instructions BASIC qui se trouvent dans les chaînes sont celles dont les symboles ont même valeur que les codes graphiques voulus, et ont été relevées dans le tableau. Ainsi, pour produire un caractère graphique de code 191, une instruction USING, dont le symbole est 191, a été placée dans la chaîne. Notons qu'il ne faut pas

mettre d'espace entre les instructions dans les chaînes, mais qu'il en faut un avant le second guillemet, pour séparer les lettres lors de l'impression. Après la pression de Enter, le BASIC va remplacer chaque instruction des lignes 20 à 50 par son symbole, puisqu'elles ne sont pas entre guillemets. USING va donc devenir 191, le Plus (+) va devenir 205, le CONT sera changé en 179, etc... La syntaxe de ces lignes est incorrecte, mais elle n'est examinée par le BASIC que lors de l'exécution du programme. Si les instructions étaient entre guillemets, le BASIC les laisserait inchangées, c'est pourquoi il faut mettre autre chose qu'un guillemet après le Egal (=). Beaucoup d'instructions ou de signes feraient par ailleurs l'affaire.

A présent, affichons notre "oui ou non" en haut de l'écran :

```
60 PRINT@ 0, CO;CU;CI;CHR$(195);CO;CU;CHR$(195);
   CN;CO;CN; : Rem le caractère 195 représente trois
   espaces (le blanc entre les mots).
```

Mais maintenant, nous devons transformer les Plus (+) en guillemets ("), faute de quoi un RUN nous donnera une SN ERROR. Le moyen le plus simple est de taper en mode commande (c'est-à-dire sans n° de ligne) une boucle transformant tous les Plus (de symbole 205) en guillemets (de code ASCII 34). Bien sûr, il faut qu'il n'y ait pas d'autres Plus dans le programme que ceux qu'on veut transformer. Comme les programmes Basic sont stockés à partir de l'adresse 17129, la boucle est :

```
FOR I=17129 TO 18000 : Rem 18000 valeur approximative
IF PEEK (I)=205 THEN POKE I,34 : NEXT I
ELSE NEXT I : Rem les octets 205 deviennent des 34
```

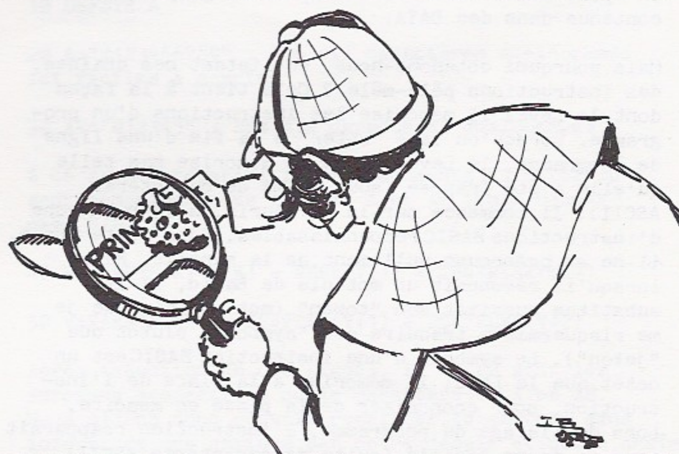
Vérifions-la soigneusement avant de taper Enter. Dès qu'elle est terminée, listons les lignes 20 à 50 : les Plus sont effectivement devenus des guillemets réglementaires, et la syntaxe est à présent correcte. Mais les chaînes contiennent toujours des CG. Il faut éviter dorénavant d'éditer ces lignes, sous peine de devoir recommencer la manipulation, après avoir remis un Plus à la place du Egal.

Le prochain volet de cet article traitera de certaines routines en langage-machine, accélérant l'affichage de certains graphiques...

Frédéric MORA

utilitaire

Détectez et modifiez vos mots-clés



Le programme MODPROG dont le listing est donné à la suite de cet article permet la recherche et le remplacement éventuel des mots "réserves" du BASIC. Exemple : remplacement de tous les PRINT par des LPRINT pour éditions sur imprimante. Il permet également la détection par numéro de ligne d'un mot-clé déterminé ou de n'importe quel caractère ASCII apparaissant dans le texte du programme. Le présent utilitaire vient donc compléter les diverses instructions REF que l'on trouve dans les systèmes d'exploitation NEWDOS et NEWDOS80 qui ne permettent que la localisation des noms de variables uniquement.

Ce programme, utilisé conjointement avec la commande MERGE du BASIC disque, peut aussi être exploité par les utilisateurs cassette, à condition qu'ils disposent de l'utilitaire réalisant cette même fonction.

LE MODE D'EMPLOI

1) Entrez sous BASIC disque et tapez le programme ci-dessous.
2) Sauvegardez le programme sous format ASCII (SAVE "MODPROG",A).
3) Tapez ou chargez le programme que vous voulez modifier.

ATTENTION : Pour ne pas entrer en conflit avec MODPROG, votre programme ne devra pas comporter de numéros de lignes inférieurs à 10. Dans le cas contraire, procéder préalablement à sa renumérotation.

4) Tapez la commande MERGE "MODPROG" pour provoquer la fusion entre votre programme et l'utilitaire de modification.
5) Répondez aux questions posées par MODPROG en vous servant du tableau de correspondance "Mot-clé/token" donné ci-contre.

DEUX EXEMPLES D'APPLICATION

a) Modification de programme

Soit à remplacer tous les "PRINT" d'un programme par des "LPRINT". Le token associé à PRINT est 178, celui associé à LPRINT est 175 en décimal (voir tableau).

- à la question "Modification du caractère de numéro de token ?", répondez : 178
- à la question "En un caractère de numéro de token ?", répondez : 175

A partir de cette étape, le programme va automatiquement remplacer les PRINT en LPRINT, en affichant pour chaque changement effectué, le numéro de la ligne concernée (Numéro de la ligne modifiée ⇒ xx).

Après cela, la question suivante est posée : "Voulez-vous détruire MODPROG ?". Si la réponse est "O", les lignes 0 à 9 sont supprimées et vous pouvez exécuter normalement le programme modifié. Dans le cas contraire (réponse différente de "O"), l'exécution de nouvelles modifications peut être entreprise.

Remarque : le remplacement d'un mot réservé par un caractère ASCII - et inversement - peut être effectué.

Une application intéressante dans les changements entre caractères ASCII est le remplacement de

	-64	DEFSNG	-154	INT	-216	OPEN	-162	STEP	-204
ABS	-217	DEFSTR	-152	KILL	-170	OR	-211	STOP	-148
AND	-210	DELETE	-182	LEFTS	-248	OUT	-160	STRINGS	-196
ASC	-246	DIM	-138	LEN	-243	PEEK	-229	STRS	-244
ATN	-228	EDIT	-157	LET	-140	POINT	-196	SYSTEM	-174
AUTO	-182	ELSE	-58, 149	LINE	-156	POKE	-177	TAB	-188
CDBL	-241	END	-128	LIST	-180	POS	-220	TAN	-227
CHRS	-247	EOF	-233	LLIST	-181	PRINT	-178	THEN	-202
CINT	-239	ERL	-194	LOAD	-167	PUT	-165	TIMES	-199
CLEAR	-184	ERR	-195	LOC	-234	RANDOM	-134	TO	-189
CLOAD	-185	ERROR	-158	LOF	-235	READ	-139	TROFF	-151
CLOSE	-166	EXP	-224	LOG	-223	REM	-147	TRON	-150
CLS	-132	FIELD	-163	LPRINT	-175	RESET	-130	USING	-191
CMD	-133	FIX	-242	LSET	-171	RESTORE	-144	USR	-193
CONT	-179	FN	-190	MEM	-200	RESUME	-159	VAL	-245
COS	-225	FOR	-129	MERGE	-168	RETURN	-146	VARPTR	-192
CSAVE	-186	FRE	-218	MIDS	-250	RIGHTS	-249	*	-207
CSNG	-240	GET	-164	MKDS	-238	RND	-222	+	-205
CVD	-232	GOSUB	-145	MKIS	-236	RSET	-172	-	-206
CVI	-230	GOTO	-141	MKSS	-237	RUN	-142	/	-208
CVS	-231	IF	-143	NAME	-169	SAVE	-173	<	-214
DATA	-136	INKEYS	-201	NEW	-187	SET	-131	=	-213
DEF	-176	INP	-219	NEXT	-135	SGN	-215	>	-212
DEFDBL	-155	INPUT	-137	NOT	-203	SIN	-226	↑	-209
DEFINT	-153	INSTR	-197	ON	-161	SOR	-221		

Codes des instructions par ordre alphabétique.

"SHIFT @" en "@". En effet, ce premier caractère, lorsqu'il est frappé par mégarde dans un "PRINT " provoque une erreur de syntaxe indétectable à la simple relecture du programme (code @ = 64, code SHIFT @ = 96).

b) Détection de mots-clés ou caractères ASCII
Soit à rechercher la présence de tous les "REM" d'un programme. La procédure est sensiblement équivalente à celle utilisée en modification et il suffit de répondre la même valeur aux deux premières questions (147 dans le cas des REM). A partir de cet instant, le programme affichera les numéros des lignes où apparaissent le token 147.

Remarque : Dans ce type d'exploitation, il peut être intéressant pour les possesseurs d'imprimante, et étant donné la rapidité d'affichage des lignes détectées, de conserver une trace écrite. Remplacez alors (manuellement !) PRINT par LPRINT dans la seconde partie des lignes 1 et 8.

ET SI VOUS DESIREZ AMELIORER LE PROGRAMME ...

Vous pouvez :

. Passer en langage machine en vue d'améliorer la rapidité d'analyse du programme.
. Faire une recherche combinée de plusieurs caractères ASCII à la fois.
. Faire une recherche et une modification opérant sur une portion du programme.
. Faire une recherche et une modification opérant directement sur un fichier disque.

Rémy Pineau

```
0 CLS : A=PEEK(16548)+PEEK(16549)*256 : FOR B=0 TO 9 : A=PEEK(A)
+PEEK(A+1)*256 : NEXT : INPUT "MODIFICATION DU CARACTERE DE NUME
RO DE TOKEN ==> ";TC : PRINT : INPUT "EN UN CARACTERE DE NUMERO
DE TOKEN ==> ";CT : GOTO4:REM14
1 TO=Z-65536 : IF PEEK(TO)=TC THEN POKE TO,CT : PRINT : IF TC<>C
T THEN PRINT "NUMERO DE LA LIGNE MODIFIEE ==> ";LN : ELSE PRINT
"NUMERO DE LIGNE OU APPARAIT LE TOKEN";CT;" ==> ";LN : GOTO2:REM
A1
2 NEXTZ : A=NP : IF PEEK(A)+PEEK(A+1)=0 THEN PRINT : PRINT "MODI
FICATIONS OU RECHERCHES TERMINEES" : PRINT ELSE IF NP<=32767 THE
N 4 ELSE NP=NP+65536 : A=NP : GOTO4:REMCE
3 INPUT "VOULEZ-VOUS DETRUIRE MODPROG (LIGNES 0-9), (O/N)";T$ :
IF T$<>"O"THEN PRINT : END ELSE PRINT : DELETE 0-9 : END:REM9A
4 L3=A : L0=A+1 : IF L3>32767 THEN L3=L3-65536 ELSE IF L0>32767
THEN L0=L0-65536:REMDF
5 NP=PEEK(L3)+PEEK(L0)*256:IF NP>32767 THEN NP=NP-65536:REMF2
6 L1=A+2 : L2=A+3 : IF L1>32767 THEN L1=L1-65536 ELSE IF L2>3276
7 THEN L2=L2-65536:REMDC
7 LN=PEEK(L1)+PEEK(L2)*256 : X=A+4 : FOR Z=X TO NP-1:IF Z>32767
THEN 1:REMAA
8 IF PEEK(Z)=TC THEN POKE Z,CT: PRINT: IF TC<>CT THEN PRINT "NUM
ERO DE LIGNE MODIFIEE ==> ";LN : ELSE PRINT "NUMERO DE LIGNE OU
APPARAIT LE TOKEN";CT;" ==> ";LN:REM38
9 GOTO 2 : Copyright TRACE et Remy Pineau    FE
```



125	150-TRON	175-LPRINT	200-MEM	225-COS
126	151-TROFF	176-DEF	201-INKEYS	226-SIN
127	152-DEFSTR	177-POKE	202-THEN	227-TAN
128-END	153-DEFINT	178-PRINT	203-NOT	228-ATN
129-FOR	154-DEFSNG	179-CONT	204-STEP	229-PEEK
130-RESET	155-DEFDBL	180-LIST	205- +	230-CVI
131-SET	156-LINE	181-LLIST	206- -	231-CVS
132-CLS	157-EDIT	182-DELETE	207-*	232-CVD
133-CMD	158-ERROR	183-AUTO	208-/	233-EOF
134-RANDOM	159-RESUME	184-CLEAR	209-↑	234-LOC
135-NEXT	160-OUT	185-CLOAD	210-AND	235-LOF
136-DATA	161-ON	186-CSAVE	211-OR	236-MKIS
137-INPUT	162-OPEN	187-NEW	212->	237-MKSS
138-DIM	163-FIELD	188-TAB	213- =	238-MKDS
139-READ	164-GET	189-TO	214-<	239-CINT
140-LET	165-PUT	190-FN	215-SGN	240-CSNG
141-GOTO	166-CLOSE	191-USING	216-INT	241-CDBL
142-RUN	167-LOAD	192-VARPTR	217-ABS	242-FIX
143-IF	168-MERGE	193-USR	218-FRE	243-LEN
144-RESTORE	169-NAME	194-ERL	219-INP	244-STRS
145-GOSUB	170-KILL	195-ERR	220-POS	245-VAL
146-RETURN	171-LSET	196-STRING$	221-SOR	246-ASC
147-REM	172-RSET	197-INSTR	222-RND	247-CHRS
148-STOP	173-SAVE	198-POINT	223-LOG	248-LEFTS
149-ELSE	174-SYSTEM	199-TIMES	224-EXP	249-RIGHTS

Codes des instructions par ordre croissant

utilitaire

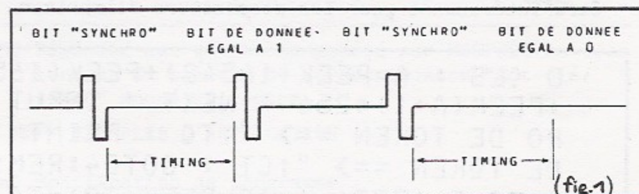
Pour lire vos cassettes « nouvelles roms »

Depuis que TANDY a modifié les ROMS du modèle 1, quelques problèmes sont rencontrés durant la lecture de certains programmes sur cassette.

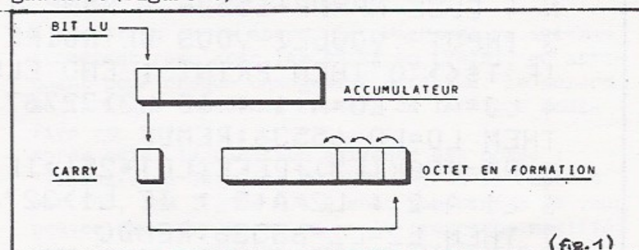
Après un survol technique sur le principe de lecture des informations issues du magnétophone, l'auteur nous donne des conseils pratiques, ainsi qu'un utilitaire permettant de remédier à ces problèmes.

PRINCIPE DE LECTURE D'UN OCTET EN PROVENANCE DE LA CASSETTE

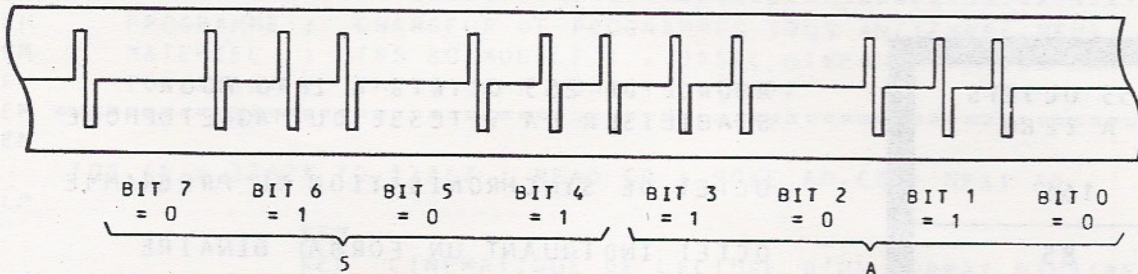
La lecture sur magnétophone se fait séquentiellement, c'est-à-dire bit par bit, au rythme de 500 bauds (500 bits par seconde). Il est donc nécessaire d'effectuer 8 lectures successives pour obtenir l'équivalent d'un octet.
D'une manière générale, l'absence de signal sur la cassette correspond à un "0" binaire alors que la présence d'une pseudo-sinusoïde correspond à un "1" binaire.
Paradoxalement, on constate qu'une absence est significative d'où l'ambiguïté de la saisir. Il a donc été admis de précéder chaque bit de donnée par un bit de synchronisation.
Ce bit de synchro est une sorte de sonnette d'alarme qui indique au micro que dans un délai donné, il doit recevoir un signal. S'il ne reçoit rien, il en déduit que c'était un bit "0", par contre, s'il reçoit quelque chose, c'est un bit "1".



La valeur ainsi recueillie ("0" ou "1") est stockée dans le bit 7 de l'accumulateur. Elle est ensuite transférée dans l'indicateur Carry puis, par une rotation circulaire, on vient la ranger dans le bit 0 de l'octet en cours de constitution (les bits précédemment lus étant décalés d'une position vers la gauche). (figure 1)



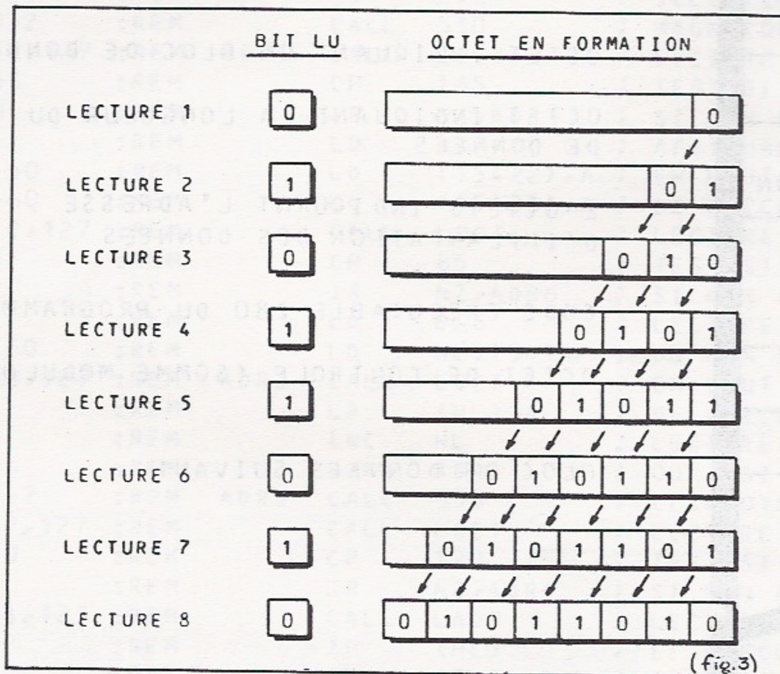
Un exemple décomposant bit par bit comment se fait l'acquisition d'un octet enregistré sur cassette, permettra de clarifier les esprits.
Supposons que l'octet à transférer en mémoire soit la lettre "Z" codée (5A) en hexadécimal et (01011010) en binaire. L'enregistrement aura physiquement l'allure de la figure 2.



(fig. 2)

A chaque lecture, la zone tampon recevant les bits de données en provenance de la cassette est décalée d'une position vers la gauche et le dernier bit recueilli est stocké à l'extrême droite de cette dernière.

A l'issue de 8 lectures consécutives (synchro + donnée) l'octet est entièrement constitué et peut être délivré pour une autre utilisation. (figure 3)



(fig. 3)

MODIFICATIONS APPORTEES PAR LE CONSTRUCTEUR
==

J'ai signalé tout à l'heure toute l'importance que présente le délai entre le bit de synchronisation et le bit de données.
Or, pour des raisons de fiabilité de lecture, TANDY a été amené à modifier ce timing dans ses nouvelles ROMS. Les octets situés aux adresses 585 et 592 sont passés respectivement de 65 à 96 et de 118 à 133. Il en résulte donc un déphasage quand on tente de lire un programme créé avec les routines de lecture des anciennes MEMS.
Ceci est une première explication aux difficultés rencontrées par les détenteurs des nouvelles ROMS (celles qui affichent MEM Size).
Il semblerait également que sur certaines cassettes vendues par TANDY le nombre d'octets réservés au nom du programme proprement dit soit de 7 au lieu de 6 (voir descriptif joint du format binaire).
Il se produit donc un décalage de un octet et la routine de lecture ne reçoit plus ce qu'elle est en droit d'attendre.
Il a été avancé également que le calcul du Checksum (octet de controle contenant la somme modulo 256 des octets constituant le bloc de données) avait été probablement modifié. Ne disposant pas des anciennes MEMS, je n'ai pu le vérifier. Mais si c'était le cas, une copie même lisible mais inchangée du programme ne ferait que repousser le problème.
Aussi le programme joint ne tient-il pas compte de cet octet de contrôle particulier.

CONSIGNES D'UTILISATION DU PROGRAMME DE CHARGEMENT
===

a) A la mise sous tension du système, protéger la partie haute de la mémoire en répondant 32628.

b) Charger ou saisir le programme joint et l'exécuter.

c) Mettre la cassette TANDY récalcitrante en position de lecture.

d) Taper alors SYSTEM puis /32628.

DEROULEMENTS POSSIBLES DES OPERATIONS
=====================================

a) Le magnétophone tourne et rien n'apparaît sur l'écran.
Régler le volume son du magnétophone jusqu'à voir apparaître quelque chose et recommencer l'opération.

b) Un "C" apparaît en haut et à droite de l'écran suivi du ready Basic.
L'octet de synchronisation a été détecté mais l'octet suivant ne correspond pas à un format binaire. Modifier légèrement le volume son et recommencer.

c) Le nom du programme apparaît à droite de l'écran puis "C" et ready.

STRUCTURE D'UN PROGRAMME EN FORMAT BINAIRE ENREGISTRE SUR CASSETTE

255 OCTETS A ZERO	AMORCE DE 255 OCTETS A ZERO POUR STABILISER LA VITESSE DU MAGNETOPHONE
165	OCTET DE SYNCHRONISATION DU PROGRAMME
85	OCTET INDIQUANT UN FORMAT BINAIRE
NOM DU PROGRAMME	NOM DU PROGRAMME SUR 6 OU 7 OCTETS SUIVANT VERSION (COMPLETE EVENTUELLEMENT PAR DES ESPACES)
60	OCTET INDIQUANT UN BLOC DE DONNEES
LBD	OCTET INDIQUANT LA LONGUEUR DU BLOC DE DONNEES
ADRESSE D'IMPLANTATION	2 OCTETS INDIQUANT L'ADRESSE D'IMPLANTATION DES DONNEES
: DONNEES :	CODE EXECUTABLE Z80 DU PROGRAMME
CHECKSUM	OCTET DE CONTROLE (SOMME MODULO 256)
60	BLOC DE DONNEES SUIVANT
: :	
: :	
CHECKSUM	DERNIER BLOC DE DONNEES
120	OCTET INDIQUANT LA FIN DU PROGRAMME
ADRESSE DE LANCEMENT	2 OCTETS INDIQUANT LE POINT D'ENTREE DU PROGRAMME

(figure 4)

Il se peut que le nom du programme fasse 7 octets et dans ce cas la détection de l'octet indiquant un bloc de données ne se fait pas.
Il y a donc lieu de modifier la ligne 2150 du programme par DATA 6,7.

d) Deux astérisques apparaissent en haut et à droite de l'écran, celui de droite clignote puis un "C" apparaît suivi de ready.
Une très légère modification du volume son est alors nécessaire.

e) Si tous les problèmes précédents sont résolus, l'exécution du programme chargé devrait intervenir à la fin de la lecture.

f) Si rien de tout cela n'y fait, il est probable qu'une partie de la cassette est déteriorée, donc illisible. Il y a toutefois encore une chance, car bien souvent, il existe un second enregistrement du programme un peu plus loin sur la cassette. Un moyen simple pour se positionner au début de celui-ci est de débrancher les 3 prises "jack" du magnétophone et d'écouter la musique binaire (BOFF). Entre les 2 enregistrements, il existe une séparation qui ne produit aucun bruit. C'est à cet endroit précis qu'il faut refaire une tentative.

RESTRICTIONS A L'UTILISATION DU PROGRAMME
===

a) Il va de soi que les programmes qui s'implanteraient au delà de l'adresse 32627 ne sont pas chargeables car ils viendraient écraser le chargeur.
b) Certains programmes (dont MICROCHESS) sont décomposés en 2 parties : la première (chargeur) ayant pour but de lire la deuxième (données) au moyen des routines de lecture de la ROM. C'est un autre cas d'impossibilité d'utilisation.

CHRISTOPHE YONNET

```
1000 REM ==========================================================
1010 REM    PROGRAMME  :  CHARGEUR DE PROGRAMMES SOUS ANCIENNES MEMS
1020 REM     MATERIEL  :  TRS 80 MODELE 1 , BASIC NIVEAU II , 16K RAM
1030 REM    COPYRIGHT  :  CHRISTOPHE YONNET  (1981)
1040 REM ==========================================================
1050 REM
1060    FOR AD = 32628 TO 32767 : READ CO : POKE AD,CO : NEXT AD
1070 REM
2000                         REM
2010                         REM  CINEMATIQUE DE LECTURE D'UN FORMAT BINAIRE
2011                         REM  ==========================================
2020                         REM
2030 DATA 205,201,1      :REM          CALL  457      ; EFFACEMENT DE L'ECRAN
2040 DATA 62,0           :REM          LD    A,0      ; SELECTION DU
2050 DATA 205,18,2       :REM          CALL  530      ; MAGNETOPHONE
2060 DATA 205,212,127    :REM  ADR1    CALL  LOCT     ; LECTURE OCTET/CASSETTE
2070 DATA 254,165        :REM          CP    165      ; TEST SI OCTET SYNCHRO
2080 DATA 32,249         :REM          JR    NZ,ADR1  ; SI NON RETOUR LECTURE
2090 DATA 62,42          :REM          LD    A,'*'    ; AFFICHAGE 2 ASTERISQUES
2100 DATA 50,62,60       :REM          LD    (15422),A ; EN HAUT ET A DROITE
2110 DATA 50,63,60       :REM          LD    (15423),A ; DE L'ECRAN
2120 DATA 205,212,127    :REM          CALL  LOCT     ; LECTURE OCTET SUIVANT
2130 DATA 254,85         :REM          CP    85       ; TEST SI FORMAT BINAIRE
2140 DATA 32,49          :REM          JR    NZ,ADR6  ; SI NON ALLER A ADR6
2150 DATA 6,6            :REM          LD    B,6      ; LECTURE ET AFFICHAGE
2160 DATA 33,54,60       :REM          LD    HL,15414 ; DU NOM DU PROGRAMME
2170 DATA 205,212,127    :REM  ADR2    CALL  LOCT     ; EN HAUT ET A DROITE
2180 DATA 119            :REM          LD    (HL),A   ; DE L'ECRAN PAR 6
2190 DATA 35             :REM          INC   HL       ; LECTURES SUCCESSIVES
2200 DATA 16,249         :REM          DJNZ  ADR2     ; OU 7 SI NECESSAIRE
2210 DATA 205,44,2       :REM  ADR3    CALL  556      ; CLIGNOTEMENT ASTERISQUE
2220 DATA 205,212,127    :REM          CALL  LOCT     ; LECTURE DU TYPE DE BLOC
2230 DATA 254,120        :REM          CP    120      ; TEST SI FIN PROGRAMME
2240 DATA 32,4           :REM          JR    NZ,ADR4  ; SI NON ALLER A ADR4
2250 DATA 205,203,127    :REM          CALL  LADR     ; LECT ADRESSE LANCEMENT
2260 DATA 233            :REM          JP    (HL)     ; ET EXECUTION
2270 DATA 254,60         :REM  ADR4    CP    60       ; TEST SI BLOC DE DONNEES
2280 DATA 32,19          :REM          JR    NZ,ADR6  ; SI NON ALLER A ADR6
2290 DATA 205,212,127    :REM          CALL  LOCT     ; LECTURE LONGUEUR BLOC
2300 DATA 71             :REM          LD    B,A      ; CHARGE DANS REGISTRE B
2310 DATA 205,203,127    :REM          CALL  LADR     ; LECT ADR IMPLANTATION
2320 DATA 205,212,127    :REM  ADR5    CALL  LOCT     ; LECTURE OCTET PROGRAMME
2330 DATA 119            :REM          LD    (HL),A   ; ET CHARGEMENT
2340 DATA 35             :REM          INC   HL       ; +1 ADRESSE IMPLANTATION
2350 DATA 16,249         :REM          DJNZ  ADR5     ; RETOUR LECTURE DONNEE
2360 DATA 205,212,127    :REM          CALL  LOCT     ; LECTURE CHECKSUM
2370 DATA 24,219         :REM          JR    ADR3     ; RETOUR LECT BLOC SUIVANT
2380 DATA 62,67          :REM  ADR6    LD    A,'C'    ; AFFICHAGE D'UN C EN HAUT
2390 DATA 50,62,60       :REM          LD    (15422),A ; ET A DROITE DE L'ECRAN
2400 DATA 195,204,6      :REM          JP    1740     ; RETOUR AU BASIC
2405                         REM
2406                         REM  ROUTINE D'ACQUISITION D'UNE ADRESSE
2407                         REM  ===================================
2408                         REM
2410 DATA 205,212,127    :REM  LADR    CALL  LOCT     ; LECTURE DE L'ADRESSE
2420 DATA 111            :REM          LD    L,A      ; D'IMPLANTATION OU
2430 DATA 205,212,127    :REM          CALL  LOCT     ; DU POINT D'ENTREE
2440 DATA 103            :REM          LD    H,A      ; DU PROGRAMME
2450 DATA 201            :REM          RET            ; SUR 2 OCTETS
2460                         REM
2470                         REM  ROUTINE D'ACQUISITION D'UN OCTET SUR CASSETTE
2480                         REM  =============================================
2490                         REM
2500 DATA 197            :REM  LOCT    PUSH  BC       ;
```

```
2510 DATA 229          :REM        PUSH  HL      ;
2520 DATA 6,8          :REM        LD    B,8     ; ACQUISITION D'UN OCTET
2530 DATA 205,224,127  :REM  ADR7  CALL  LBIT    ; PAR 8 LECTURES
2540 DATA 16,251       :REM        DJNZ  ADR7    ; SUCCESSIVES D'UN BIT
2550 DATA 225          :REM        POP   HL      ;
2560 DATA 193          :REM        POP   BC      ;
2570 DATA 201          :REM        RET           ;
2580                    REM
2590                    REM  ROUTINE D'ACQUISITION D'UN BIT SUR CASSETTE
2600                    REM  ===========================================
2610                    REM
2620 DATA 197          :REM  LBIT  PUSH  BC      ;
2630 DATA 245          :REM        PUSH  AF      ;
2640 DATA 219,255      :REM  ADR8  IN    A,255   ; BOUCLE D'ATTENTE
2650 DATA 23           :REM        RLA           ; DU BIT DE
2660 DATA 48,251       :REM        JR    NC,ADR8 ; SYNCHRONISATION
2670 DATA 6,65         :REM        LD    B,65    ; PREMIER TIMING MODIFIE
2680 DATA 16,254       :REM  ADR9  DJNZ  ADR9    ; DANS NOUVELLES ROMS
2690 DATA 205,30,2     :REM        CALL  542     ; RAZ TEMPON DE LECTURE
2700 DATA 6,118        :REM        LD    B,118   ; DEUXIEME TIMING MODIFIE
2710 DATA 16,254       :REM  ADR10 DJNZ  ADR10   ; DANS NOUVELLES ROMS
2720 DATA 219,255      :REM        IN    A,255   ; LECTURE BIT DE DONNEE
2730 DATA 71           :REM        LD    B,A     ;
2740 DATA 241          :REM        POP   AF      ;
2750 DATA 203,16       :REM        RL    B       ; BIT LU DANS CARRY
2760 DATA 23           :REM        RLA           ; PUIS DANS L'ACCUMULATEUR
2770 DATA 245          :REM        PUSH  AF      ;
2780 DATA 205,30,2     :REM        CALL  542     ; RAZ TEMPON DE LECTURE
2790 DATA 241          :REM        POP   AF      ;
2800 DATA 193          :REM        POP   BC      ;
2810 DATA 201          :REM        RET           ;
```

initiation

Et bien compilez maintenant

Vous avez probablement souffert de la lenteur excessive de certains programmes BASIC. Ce n'est pas toujours très gênant, mais dans bien des cas cela devient très pénalisant. Prenez l'exemple de ce labyrinthe qui se dessine avec une sage lenteur: c'est énervant à force, et ça déconcentre pour la suite du jeu ! Avec un compilateur BASIC, tel que celui de Microsoft, une à deux secondes suffiront pour réaliser le même travail. Tentant, non ?

Il ne s'agira pas dans cet article, d'expliquer ce qu'est un compilateur. Nous nous bornerons tout simplement à dire que c'est un logiciel qui transforme un programme écrit en langage évolué (BASIC) en une sorte de langage machine, directement assimilable par le microprocesseur. En supprimant les intermédiaires (interprêteur entre autre), il en résulte un gain de temps important dans l'exécution du programme, qui peut atteindre un rapport de 20 ou 30. Ce n'est pas à négliger !
En réalité, il s'avère que beaucoup d'utilisateurs possèdent le compilateur BASCOM, mais que peu d'entre eux osent l'utiliser, ne serait-ce que parce que cela leur semble très compliqué, ou parce que la documentation est en anglais. Nous espèrons que cet article les aidera à faire le premier pas.

Les lignes qui suivent décrivent la succession des opérations nécessaires pour compiler un petit programme BASIC. Pour ne pas tricher, nous avons dirigé tous les messages écran sur l'imprimante, au moyen de la commande ROUTE de NEWDOS80. Seules, les portions de textes frappées au clavier sont soulignées sur le listing et quelques "bulles" qui passaient par là se sont accrochées à certaines lignes ...

Le compilateur BASIC de Microsoft est composé de quatre programmes : BASCOM (le compilateur proprement dit), L80 (l'éditeur de liens qui permet l'union de différents programmes), BASLIB (la bibliothèque contenant les routines usuelles du BASIC) et BRUN (ou Basic Run, l'exécuteur des programmes compilés).

Le programme utilisé pour notre exemple, créé préalablement avec le BASIC normal, est appelé CARRE/BAS. Il calcule les carrés des dix premiers nombres et exécute un test de vitesse d'une boucle dans laquelle a été glissé un "J=1". Nous verrons pourquoi un peu plus tard. La date et l'heure en début et en fin de boucle est affichée.
Ce programme est lancé sous l'interprêteur BASIC par une commande RUN. Après s'être assuré qu'il fonctionne correctement, il est ensuite sauvé sur

```
NEWDOS/80 READY
BASIC LOAD "CARRE/BAS"
DISK BASIC. RADIO SHACK'S ROM ENHANCED WITH
APPARAT'S NEWDOS80 EXTENDED AND DISK FEATURES.
>READY
>LIST
10   Exemple de programme BASIC a compiler
20 CLS
30 PRINT TAB(15) ; "-- CARRES DES 10 PREMIERS NOMBRES --"
40 PRINT
50 FOR NOMBRE% =1 TO 10
60 PRINT NOMBRE% ; "au carre donne" ; NOMBRE% ↑ 2
70 NEXT : PRINT
80 INPUT "encore une fois" ; REP$
90 IF REP$ = "OUI" OR REP$="O" THEN 10
100 PRINT "*** TEST DE VITESSE ***"
110 PRINT "Heure avant la boucle : " ; TIME$
120 FOR N%=1 TO 10000 : J=1 : NEXT
130 PRINT "heure apres la boucle : " ; TIME$
READY
>RUN                    -- CARRES DES 10 PREMIERS NOMBRES --

 1 au carre donne 1
 2 au carre donne 4
 3 au carre donne 9
 4 au carre donne 16
 5 au carre donne 25
 6 au carre donne 36
 7 au carre donne 49
 8 au carre donne 64
 9 au carre donne 81
 10 au carre donne 100

encore une fois? N
*** TEST DE VITESSE ***
Heure avant la boucle : 11/11/11 11:32:26
heure apres la boucle : 11/11/11 11:33:25
READY
>save"CARRE/BAS:0",A
READY
>cmd"s"

NEWDOS/80 READY
BASCOM
*CARRE:0,CARRE:0=CARRE:0 -S

00000 FATAL ERROR(S)
12635 BYTES FREE

NEWDOS/80 READY
L80

LINK-80  3.41  28-DEC-80  COPYRIGHT 1979,80 (C) MICROSOFT

*CARRE:0 -S
DATA     8A00     8BCC     < 46D>

 BASLIB RQUEST
-$4.0   8A31    -$AC     8A75    -$BEXC   8A70
-$CISA  8A6D    -$CLS    8A34    -$END    8B18
```

drôle de date...

Δ = 59 secondes...

ne pas oublier le "A" !

objet — liste — source

Quelle chance !

I

II

```
-$EQSA  8ABF     -$FASA  8AEC     -$INOA  8A9D
-$INI   8A20     -$IPUA  8AA4     -$IPUB  8AAC
-$PROA  8B09     -$PV1C  8A61     -$PV1D  8B0F
-$PV2A  8A78     -$PV2D  8B15     -$TABB  8A3D
-$TIM   8B12
  19 UNDEFINED GLOBAL(S)
34865 BYTES FREE

*BASLIB-S

*-U
DATA     8A00     8D72    <  882>

 BASLIB RQUEST
29385 BYTES FREE

*CARRE:0-N-E

DATA     8A00     8D72    <  882>

 BASLIB RQUEST
29385 BYTES FREE
(8A1C   8D72)

NEWDOS/80 READY
DIR
DRIVE   0  NEWDOS80   02/02/82   39 TRKS   20 FDES   27 GRANS

 CARRE/LST       CARRE/REL       CARRE/CHN       CARRE/BAS

NEWDOS/80 READY
BRUN CARRE
               -- CARRES DES 10 PREMIERS NOMBRES --

 1 au carre donne 1
 2 au carre donne 4
 3 au carre donne 9.00003
 4 au carre donne 16
 5 au carre donne 25
 6 au carre donne 36.0001
 7 au carre donne 49.0003
 8 au carre donne 64
 9 au carre donne 81
 10 au carre donne 100

encore une fois? N
*** TEST DE VITESSE ***
Heure avant la boucle : 00/00/00 00:32:58
heure apres la boucle : 00/00/00 00:33:02
NEWDOS/80 READY
LIST CARRE/LST

BASCOM 5.23 - COPYRIGHT 1979, 80 (C) BY MICROSOFT - 13223 BYTES FREE
 0014 0007       10 ' EXEMPLE DE PROGRAMME BASIC A COMPILER
       ** 0014'          CALL      $4.0
       ** 0017'L00010:
 0017 0007       20 CLS
       ** 0017'L00020: CALL      $CLS
 001A 0007       30 PRINT TAB(15);"-- CARRES DES 10 PREMIERS NOMBRES --"
       ** 001A'L00030: CALL      $PROA
```

```
   ** 001D'           LD         HL,000F          } tabulation
   ** 0020'           CALL       $TABB
   ** 0023'           LD         HL,<CONST>       ]
   ** 0026'           CALL       $PV1D            ]
   ** 0029'           LD         HL,<CONST>       ]
   ** 002C'           CALL       $PV2D            ]
002F 0007       40 PRINT
   ** 002F'L00040:    CALL       $PROA
   ** 0032'           LD         HL,<CONST>
   ** 0035'           CALL       $PV2D
0038 0007       50 FOR NOMBRE% =1 TO 10
   ** 0038'L00050:    LD         HL,0001
   ** 003B'           LD         (NO%),HL
   ** 003E'I00000:

003E 0009       60 PRINT NOMBRE% ; "AU CARRE DONNE" ; NOMBRE% ↑ 2
   ** 003E'L00060:    CALL       $PROA
   ** 0041'           LD         HL,(NO%)         ]
   ** 0044'           CALL       $PV1C            ]
   ** 0047'           LD         HL,<CONST>       ]
   ** 004A'           CALL       $PV1D            ]
   ** 004D'           LD         HL,(NO%)
   ** 0050'           CALL       $CISA
   ** 0053'           CALL       $BEXC
   ** 0056'           DW         <CONST>
   ** 0058'           LD         HL,$AC%
   ** 005B'           CALL       $PV2A
005E 0009       70 NEXT : PRINT
   ** 005E'L00070:    LD         HL,(NO%)         ]
   ** 0061'           INC        HL               ] +1 sur NO%
   ** 0062'           LD         (NO%),HL         ]
   ** 0065'           LD         HL,(NO%)         ]
   ** 0068'           LD         DE,FFF5
   ** 006B'           LD         A,H
   ** 006C'           RLA
   ** 006D'           JR         C,02
   ** 006F'           ADD        HL,DE
   ** 0070'           ADD        HL,HL
   ** 0071'           JP         C,I00000
   ** 0074'           CALL       $PROA
   ** 0077'           LD         HL,<CONST>
   ** 007A'           CALL       $PV2D

007D 0009       80 INPUT "ENCORE UNE FOIS" ; REP$
   ** 007D'L00080:    LD         HL,<CONST>
   ** 0080'           CALL       $INOA
   ** 0083'           DB         00
   ** 0084'           JP         Z,I00001
   ** 0087'           CALL       $IPUA
   ** 008A'           DB         01
   ** 008B'           DB         07
   ** 008C'           LD         HL,RE$
   ** 008F'           CALL       $IPUB
   ** 0092'I00001:

0092 000C       90 IF REP$ = "OUI" OR REP$="O" THEN 10
   ** 0092'L00090:    LD         DE,RE$
   ** 0095'           LD         HL,<CONST>
   ** 0098'           CALL       $EQSA
```

Seuls les 2 premiers caractères sont conservés

```
          ** 009B'            LD        DE,RE$
          ** 009E'            PUSH      HL
          ** 009F'            LD        HL,<CONST>
          ** 00A2'            CALL      $EQSA
          ** 00A5'            POP       DE
          ** 00A6'            LD        A,H
          ** 00A7'            OR        D
          ** 00A8'            LD        H,A
          ** 00A9'            LD        A,L
          ** 00AA'            OR        E
          ** 00AB'            LD        L,A
          ** 00AC'            LD        A,H
          ** 00AD'            OR        L
          ** 00AE'            JP        NZ,L00010

00B1 000C          100 PRINT "*** TEST DE VITESSE ***"
          ** 00B1'L00100: CALL      $PROA
          ** 00B4'            LD        HL,<CONST>
          ** 00B7'            CALL      $PV2D

00BA 000C          110 PRINT "HEURE AVANT LA BOUCLE : " ; TIME$
          ** 00BA'L00110: CALL      $PROA
          ** 00BD'            LD        HL,<CONST>
          ** 00C0'            CALL      $PV1D
          ** 00C3'            CALL      $TIM
          ** 00C6'            CALL      $PV2D

00C9 000C          120 FOR N%=1 TO 10000 : J=1 : NEXT
          ** 00C9'L00120: LD        HL,0001
          ** 00CC'            LD        (N%),HL
          ** 00CF'I00003:
          ** 00CF'            CALL      $FASA
          ** 00D2'            DW        J!
          ** 00D4'            DW        <CONST>
          ** 00D6'            LD        HL,(N%)
          ** 00D9'            INC       HL
          ** 00DA'            LD        (N%),HL
          ** 00DD'            LD        HL,(N%)
          ** 00E0'            LD        DE,D8EF
          ** 00E3'            LD        A,H
          ** 00E4'            RLA
          ** 00E5'            JR        C,02
          ** 00E7'            ADD       HL,DE
          ** 00E8'            ADD       HL,HL
          ** 00E9'            JP        C,I00003

00EC 0012          130 PRINT "HEURE APRES LA BOUCLE : " ; TIME$
          ** 00EC'L00130: CALL      $PROA
          ** 00EF'            LD        HL,<CONST>
          ** 00F2'            CALL      $PV1D
          ** 00F5'            CALL      $TIM
          ** 00F8'            CALL      $PV2D

00FB 0012
          ** 00FB'            CALL      $END
01AF 001C

00000 FATAL ERROR(S)
12635 BYTES FREE
```

Le "J=1" reste dans la boucle

```
NEWDOS/80 READY
DIR A
DRIVE   0  NEWDOS80  02/02/82   39 TRKS   20 FDES   27 GRANS

                     EOF    LRL   RECS GRANS EXTS  SIUEC....UAL
CARRE/LST           17/191  256    18    4    2   ..U........0
CARRE/REL            2/233  256     3    1    1   ..U........0
CARRE/CHN            3/134  256     4    1    1   ..U........0
CARRE/BAS            1/185  256     2    1    1   ..U........0

NEWDOS/80 READY
```

disque sous le nom CARRE/BAS. Pour satisfaire le compilateur, il est nécessaire de le sauver en format ASCII (d'où le ",A" à la fin de la commande).

Un retour au DOS est alors effectué (CMD "S") et le compilateur BASCOM est appelé. C'est la phase I. Comme il est peu bavard, il se contente d'afficher un "*" auquel il suffit de répondre par :

CARRE:0,CARRE:0=CARRE:0

ce qui signifie :
le nom du programme objet (issu de la compilation) se nommera CARRE et sera placé sur le disque 0, le nom du fichier listé (liste de compilation) se nommera CARRE et sera placé sur le disque 0, et le nom du programme source appelé CARRE se trouve sur le disque 0 ...
Si l'on veut aller plus vite, il est possible de passer ces paramètres au moment de l'appel du compilateur :

BASCOM CARRE:0,CARRE:0=CARRE:0

En réalité, les trois noms sont distingués par une extension implicite (qu'il n'est pas nécessaire de spécifier) : /REL pour l'objet, /LST pour la liste et /BAS pour le source. Si l'on ne désirait pas de fichier liste, il aurait suffit de frapper :

CARRE:0=CARRE:0 -S

et si l'on veut une liste mais pas d'objet :

,CARRE:0=CARRE:0

Le ":0" n'est évidemment pas obligatoire et le cas échéant, le système se débrouille tout seul.
Il est aussi possible de faire :

=CARRE

et dans ce cas, aucune génération de fichiers objet ou liste ne sera produite. Cela peut permettre de s'assurer avant toute chose, qu'il n'y a pas d'erreurs de compilation.
Dans le cas de notre exemple, la compilation se termine bien puisqu'aucune erreur fatale (certaines ne le sont pas et n'empèchent pas "l'édition de liens" que nous verrons par la suite) n'est signalée. Un autre message nous indique que notre programme aurait pu être plus long de 12635 octets.

Après la compilation, notre programme objet n'est toujours pas exécutable mais se trouve sous une forme intermédiaire (relatif ou relogeable). Il faut "éditer les liens" en appelant le "linker" : L80. C'est la phase II.
Après l'astérisque, frappons le nom de notre programme objet : CARRE:0.-S

Après un bref instant, le "linker" nous signale que ce programme fait référence à des routines externes, et il en dresse la liste (il y en a 19).
Il est donc nécessaire de satisfaire ces références

en donnant le nom d'un fichier spécial contenant la librairie des routines utilisées en BASIC. Ce fichier s'appelle BASLIB. Afin de ne pas charger la totalité de la librairie (qui est énorme), il faut faire suivre le nom de l'indicateur "-S", afin que seules les routines SELECTIONNEES soient appelées en mémoire. Cette opération prend quelques minutes, aussi soyez patients !

La commande "-U" nous indique cette fois que toutes les références ont été satisfaites.
Il ne reste plus qu'à sauver le programme en frappant :

CARRE:0-N-E

ce qui signifie : sauver sur le disque 0 le programme exécutable sous le nom "CARRE" (extension implicite /CHN), et retourner au DOS (-N voulant dire : sauver le programme sous le Nom qui précéde, et -E : Exit).

La commande DIRectory nous indique que quatre fichiers figurent maintenant sur notre disquette.
Pour exécuter notre programme compilé, il suffit maintenant d'invoquer l'exécuteur (BRUN) en lui spécifiant le nom du fichier contenant le programme à lancer (extension implicite /CHN) :

BRUN CARRE

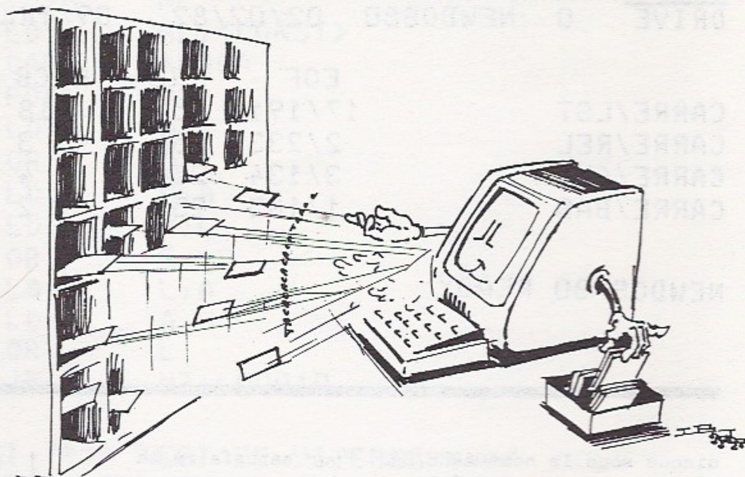
Comme nous pouvons le constater, le calcul d'arrondi est différent de celui employé par l'interprêteur (9.00003 au lieu de 9, par exemple). Le temps d'exécution est différent lui aussi ... puisqu'il passe de 59 secondes en interprété à 4 secondes en compilé.

Enfin pour les petits curieux qui veulent en savoir plus, nous donnons la liste de compilation qui montre comment chaque commande BASIC est traduite en code assembleur par le compilateur. Cela nous permet de voit entre autre, que le "J=1" reste dans la boucle (certains compilateurs l'auraient retiré dans cet exemple précis). Sans cela, le temps d'exécution de la boucle serait passé de 4 à 2 secondes ... Dommage !
Pour terminer, nous donnons la taille relative de chaque fichier (commande DIR A).

Alors, les hésitants ? Etait-ce si difficile que cela ?

J.P. du Tillet

Un tri efficace et simple

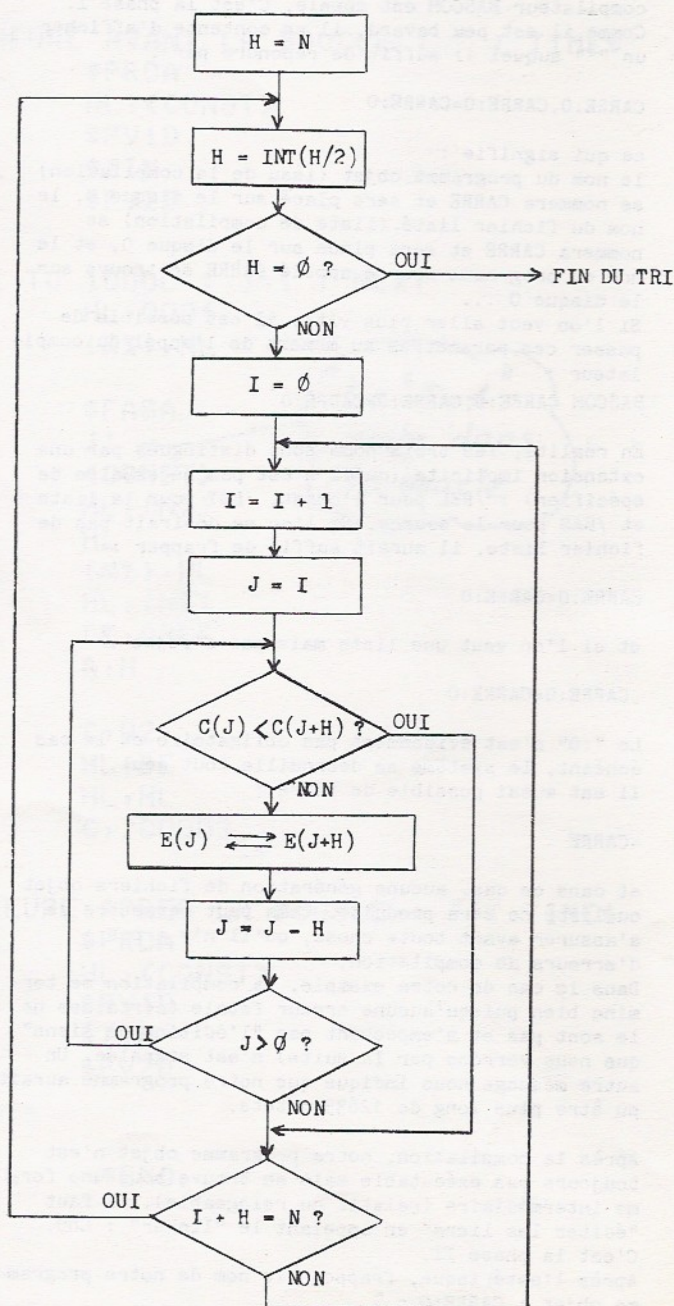


La nécessité du tri apparaît lorsqu'on aborde la gestion des données. Il s'impose pour tirer parti de la souplesse qu'offrent les fichiers à accès direct, notamment lors d'une consultation sélective où l'on pourra pratiquer la recherche dichotomique dans une table de clés triée.

L'imagination fertile des théoriciens a produit de nombreuses méthodes permettant d'ordonner une suite de données. Mais leur multiplicité soulève le problème du choix pour le programmeur, car aucune ne rallie tous les suffrages. En règle générale, plus la méthode est simple à comprendre et donc à programmer, moins elle est performante. Le temps d'exécution croît de manière exponentielle avec le nombre d'éléments à trier. En BASIC interprété, dès que ce nombre devient un peu important (disons au delà de 50 ou 100 suivant la méthode employée), le tri s'éternise et l'opérateur scrute désespérément son écran figé pour de longues minutes, voire des heures, en rêvant peut-être à une méthode magique qui effectuerait ce travail à la vitesse ... d'un ordinateur. Il n'y a sans doute pas de place pour la magie en informatique, mais quelques 170 octets d'une routine en langage machine peuvent néanmoins vous faire passer du rêve à la réalité.

L'algorithme retenu pour cette routine est celui de SHELL-METZNER qui allie une bonne efficacité à une relative simplicité de programmation.
Rappelons que la méthode SHELL est comparable à un procédé RIPPLE SORT dans lequel les comparaisons successives s'effectueraient, non plus entre éléments consécutifs, mais entre éléments distants d'un pas H. Comme dans le procédé précité, la liste des éléments est parcourue autant de fois que nécessaire jusqu'à ce qu'il n'y ait plus de permutations à effectuer. Ensuite le pas H, pris au départ égal à la moitié du nombre N des éléments à trier, est divisé par 2 et un nouveau cycle recommence. Ainsi de suite jusqu'à ce que H (qui varie par valeurs entières, bien sûr) devienne égal à 0, ce qui marque évidemment la fin du tri. Le dernier cycle (où H=1) est donc analogue au cycle RIPPLE SORT, mais porte sur une liste déjà presque ordonnée par les cycles précédents qui ont été effectués avec de grands pas. Il en résulte une notable diminution du nombre total des comparaisons, aussi bien que des inversions, pour arriver au tri complet.

L'amélioration de METZNER concerne le cycle des explorations de la liste. Après chaque permutation, une comparaison est faite avec le précédent élément distant du même pas H. Une nouvelle permutation est alors effectuée le cas échéant, et ainsi de suite



jusqu'à ce que la comparaison ne donne plus lieu à permutation ou que le début de la table soit dépassé. Alors l'exploration reprend là où elle avait été interrompue. Ainsi un seul passage suffit pour chaque pas H, ce qui réduit le nombre des comparaisons. D'autre part, il n'y a plus besoin de positionner un témoin de permutation.

Pour un tri en ordre croissant, l'algorithme de SHELL-METZNER peut être représenté par l'organigramme ci-après, où :

- I et J sont les indices courants des tableaux. J a une valeur différente de I lorsqu'il y a retour en arrière (cycle METZNER). La confusion de ces deux variables augmenterait un peu le nombre des comparaisons, le saut de retour à l'exploration étant remplacé par un pas à pas superflu.

- C(J) et C(J+H) sont les clés des éléments à trier E(J) et E(J+H), et sont donc normalement incluses dans ces dernières. Si tel n'était pas le cas, il faudrait permuter également les clés.

Dans la pratique, les fichiers sont généralement trop volumineux pour tenir dans la mémoire centrale. On utilise alors une table d'index dont chaque élément est constitué d'une clé de recherche (par exemple, le nom pour un fichier d'adresses) associée au numéro de l'enregistrement correspondant. Si l'on prend la précaution de toujours coder ce numéro sur un même nombre de caractères ASCII (les caractères manquants des premiers numéros étant remplacés par des zéros ou des espaces), et de les placer immédiatement à la suite des caractères de la clé, on peut alors confondre clé de tri et élément à trier. Nous obtiendrons ainsi un tri stable malgré l'utilisation de la méthode SHELL-METZNER.

Dans un souci de simplicité, seul le tri de chaînes de caractères a été envisagé, ce qui couvre les besoins les plus courants. Les éléments seront classiquement définis par un tableau à une seule dimension, et l'élément d'ordre 0 ne devra pas être utilisé. Pour le fonctionnement de cette routine nous avons délibérément choisi d'effectuer les permutations, non pas sur les éléments eux-mêmes, mais sur leurs pointeurs en zone tableau. Le programme ne s'en trouve certes pas simplifié, mais la vitesse est augmentée et devient indépendante de la longueur des éléments. D'autre part, il devient possible de trier des éléments de longueur inégale.

La mise en oeuvre est particulièrement simple. Si ES est le nom de variable affecté aux éléments à trier, la routine sera appelée par l'instruction U=USRn(VARPTR(ES(0))). En tête, un petit programme d'exploration détermine automatiquement le nombre N des éléments à trier. Celui-ci est limité, soit par la dimension donnée au tableau, soit par le rang du dernier élément défini s'il lui est inférieur. Ceci permet à l'usager de dimensionner à priori le tableau dans le programme sans se préoccuper du nombre réel d'éléments de la table.

Certaines variantes peuvent être envisagées. Si les éléments à trier ont toujours la même longueur (cas général des tables d'index), on peut supprimer les 17 octets des instructions repérées par un astérisque. Pour trier en ordre décroissant, intercaler l'instruction assembleur SCF (code 37H) entre les lignes 900 et 910. N'utilisant que des adressages relatifs, cette routine peut être relogée n'importe où en mémoire. Toutefois n'oubliez pas de la protéger en répondant à la question MEMORY SIZE ?

Pour ceux qui restent allergiques à l'assembleur (si, si, il y en a), le programme ci-dessous leur permettra quand même de bénéficier des avantages du langage machine. Destiné à être incorporé dans un programme principal de gestions de fichiers en BASIC, il implante la routine de tri dans l'espace libre de la mémoire centrale, juste au-dessous de la pile. De ce fait, la routine est autoprotégée à condition toutefois qu'il reste au moins 400 octets disponibles pendant l'exécution du programme.

```
10 '-----------------------------------------------------------'   F6
20 '                                                          '   EC
30 '  DETERMINATION DE VALEURS ALEATOIRES                     '   E2
40 '                                                          '   D8
50 INPUT "NOMBRE D'ELEMENTS" ; N                              '   ED
60 CLEAR N*14                                                 '   8A
70 N=FRE(E$)/14 : DIM E$(N)                                   '   02
80 X=1E6                                                      '   D7
90 FOR I=1 TO N                                               '   CB
100     E$(I)=MID$(STR$(INT(RND(0)*X)),2)                     '   27
110     IF LEN(E$(I))<6 THEN E$(I)="0"+E$(I) : GOTO 110       '   71
120 NEXT                                                      '   01
130 PRINT TIME$              PRINT "DEBUT" EN VERSION K7      '   05
140 GOSUB 1000               APPEL ROUTINE DE TRI             '   22
150 PRINT TIME$            PRINT "FIN" EN VERSION K7          '   F1
160 END                                                       '   E0
1000 '---------------------------------------------------------'   15
1010 '                                                        '   0B
1020 ' ROUTINE DE TRI BINAIRE                                 '   01
1030 ' COPYRIGHT TRACE ET ROGER BUVAT                         '   F6
1040 '                                                        '   EC
1050 A%=PEEK(16544)+256*(PEEK(16545)-2)-65536 : DEFUSR=A%     '   EB
1060 FOR I=0 TO 169 : READ J : POKE A%+I,J : NEXT             '   FF
1070 V=USR(VARPTR(E$(0))) : RETURN                            '   CB
1080 '                                                        '   C4
1090 DATA 205,127,10,43,70,43,78,35,35,229,17,0,0,11:         '   EF
```

```
1100 DATA 35,35,35,126,183,40,6,19,11,120,177,32,243,225: '  16
1110 DATA 66,75,120,203,63,71,121,31,79,176,200:            C3
1120 DATA 229,221,225,229,9,9,9,229,253,225,33,0,0,35,213:' CB
1130 DATA 17,3,0,221,25,253,25,221,229,253,229,229,43,217:' C5
1140 DATA 253,102,2,253,110,1,221,86,2,221,94,1,221,70,0:  ' 0E
1150 DATA 253,78,0,197,121,184,32,1,65,213,229:           ' C5
1160 DATA 26,190,56,6,32,24,19,35,16,246,217,225,225,225: ' CE
1170 DATA 225,253,225,221,225,209:                           37
1180 DATA 229,9,237,82,225,32,187,225,24,162:             ' 00
1190 DATA 225,221,116,2,221,117,1,209,253,114,2,253,115,1:' A2
1200 DATA 193,253,112,0,221,113,0,217,237,66,56,214:      ' AD
1210 DATA 235,221,229,253,225,221,229,225:                ' 82
1220 DATA 237,66,237,66,237,66,229,221,225,235,24,153:   '  16
```

Les utilisateurs de Level II cassette remplaceront l'affichage du temps (TIME$) par l'affichage des mots "DEBUT" et "FIN". Ils devront également remplacer la ligne 1050 par les 2 lignes :

```
1050 A%=PEEK(16544)+256*(PEEK(16545)-2)                   A8
1055 POKE 16526,(A% AND 255) : POKE 16527,A%/256          E8
```

Sur le plan des performances, la vitesse de tri est proprement stupéfiante.

Pour la mesurer, le petit programme placé en tête du listing créera des éléments aléatoires de 6 caractères numériques.

Si vous avez encore la nostalgie du tri en BASIC, comparez les temps avec cette programmation de l'algorithme de SHELL-METZNER :

```
1000 '-----------------------------------------------------  15
1010 '                                                      ' 0B
1020 ' TRI BASIC (METHODE SHELL/METZNER)                    ' 01
1030 '                                                      ' F6
1040 H=N                                                    ' 81
1050 H=INT(H/2) : IF H=0 RETURN                             ' AA
1060 FOR I=1 TO N-H : J=I                                   ' 45
1070 IF E$(J)<E$(J+H) THEN 1100                             ' C0
1080   E$=E$(J) : E$(J)=E$(J+H) : E$(J+H)=E$                ' C5
1090   J=J-H : IF J>0 THEN 1070                             ' 92
1100 NEXT : GOTO 1050                                       ' 90
```



en la substituant aux lignes 1000 à 1220 du programme.

Roger Buvat

```
        00010 ; ROUTINE DE TRI SHELL-METZNER
        00020 ;
        00030 ; AUTEUR : ROGER BUVAT
        00040 ; COPYRIGHT "TRACE" ET L'AUTEUR
        00050 ;
        00060 ; TRI EN ORDRE CROISSANT
        00070 ; DE CHAINES DE CARACTERES ALPHANUMERIQUES
        00080 ; EN TABLEAU A UNE SEULE DIMENSION
        00090 ; ======================================
        00100           ORG     0FF00H
        00110
        00120 ; ACQUISITION DES PARAMETRES
        00130 ; --------------------------
        00140
CD7F0A  00150           CALL    0A7FH           ; VARPTR DE E$(0) DANS HL
2B      00160           DEC     HL
46      00170           LD      B,(HL)
2B      00180           DEC     HL
4E      00190           LD      C,(HL)
```

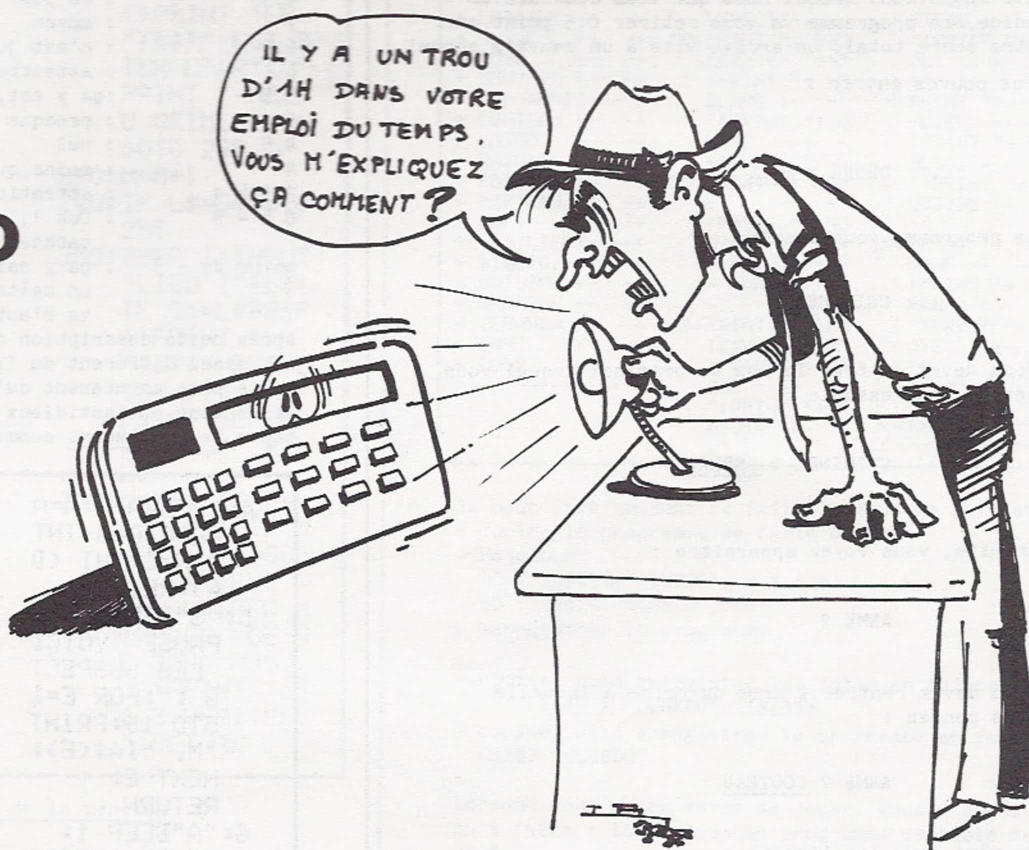
```
23        00200          INC    HL
23        00210          INC    HL
E5        00220          PUSH   HL
110000    00230          LD     DE,0000
0B        00240          DEC    BC
23        00250 EXPL1    INC    HL
23        00260          INC    HL
23        00270          INC    HL
7E        00280          LD     A,(HL)
B7        00290          OR     A
2806      00300          JR     Z,EXPL2          ; SI ELEMENT NON DEFINI
13        00310          INC    DE
0B        00320          DEC    BC
78        00330          LD     A,B
B1        00340          OR     C
20F3      00350          JR     NZ,EXPL1         ; RETOUR DE BOUCLE
E1        00360 EXPL2    POP    HL
          00370
          00380 ; ROUTINE DE TRI
          00390 ; --------------
          00400
42        00410          LD     B,D              ; (N DANS DE)
4B        00420          LD     C,E              ; H=N (H DANS BC)
78        00430 TRI1     LD     A,B
CB3F      00440          SRL    A
47        00450          LD     B,A
79        00460          LD     A,C
1F        00470          RRA
4F        00480          LD     C,A              ; H=INT(H/2) (H DANS BC)
B0        00490          OR     B
C8        00500          RET    Z                ; SI FIN DU TRI
E5        00510          PUSH   HL
DDE1      00520          POP    IX
E5        00530          PUSH   HL               ; SAUVE PTR DE TABLEAU
09        00540          ADD    HL,BC
09        00550          ADD    HL,BC
09        00560          ADD    HL,BC
E5        00570          PUSH   HL
FDE1      00580          POP    IY
210000    00590          LD     HL,0000          ; I=0
23        00600 TRI2     INC    HL               ; I=I+1
D5        00610          PUSH   DE               ; SAUVE N
110300    00620          LD     DE,0003
DD19      00630          ADD    IX,DE
FD19      00640          ADD    IY,DE
DDE5      00650          PUSH   IX               ; SAUVE PTR DE E$(I)
FDE5      00660          PUSH   IY               ; SAUVE PTR DE E$(I+H)
E5        00670          PUSH   HL               ; SAUVE I
2B        00680          DEC    HL               ; J-1=I-1
D9        00690 TRI3     EXX
FD6602    00700          LD     H,(IY+02)
FD6E01    00710          LD     L,(IY+01)
DD5602    00720          LD     D,(IX+02)
DD5E01    00730          LD     E,(IX+01)
DD4600    00740          LD     B,(IX+00)
          00750
          00760 ; * EN CAS DE LONGUEUR VARIABLE DES ELEMENTS A TRIER
          00770
FD4E00    00780          LD     C,(IY+00)
C5        00790          PUSH   BC
79        00800          LD     A,C
```

```
B8                 CP      B
2001               JR      NZ,$+3
41                 LD      B,C

      ; SUITE DE LA ROUTINE DE TRI

D5                 PUSH    DE              ; SAUVE ADRS DE E$(J)
E5                 PUSH    HL              ; SAUVE ADRS DE E$(J+H)
1A         TRI4    LD      A,(DE)
BE                 CP      (HL)
3806               JR      C,TRI5          ; SI E$(J) < E$(J+H)
2018               JR      NZ,PERMUT       ; SI E$(J) > E$(J+H)
13                 INC     DE
23                 INC     HL
10FE               DJNZ    TRI4
D9         TRI5    EXX
E1                 POP     HL
E1                 POP     HL
E1                 POP     HL              ; * SI LONGUEUR VARIABLE
E1         TRI6    POP     HL              ; RESTAURE I
FDE1               POP     IY              ; RESTAURE PTR DE E$(I+H)
DDE1               POP     IX              ; RESTAURE PTR DE E$(I)
D1                 POP     DE              ; RESTAURE N
E5                 PUSH    HL
09                 ADD     HL,BC           ; I+H
ED52               SBC     HL,DE           ; I+H-N
E1                 POP     HL
20BB               JR      NZ,TRI2         ; RETOUR DE BOUCLE
E1                 POP     HL              ; RESTAURE PTR DE TABLEAU
18A2               JR      TRI1            ; SI I+H-N=0
E1         PERMUT  POP     HL              ; RESTAURE ADRS DE E$(J+H)
DD7402             LD      (IX+02),H
DD7501             LD      (IX+01),L
D1                 POP     DE
FD7202             LD      (IY+02),D
FD7301             LD      (IY+01),E
C1                 POP     BC              ; * SI LONGUEUR VARIABLE
FD7000             LD      (IY+00),B       ; * SI LONGUEUR VARIABLE
DD7100             LD      (IX+00),C       ; * SI LONGUEUR VARIABLE
D9                 EXX
ED42               SBC     HL,BC           ; J-1-H
38D6               JR      C,TRI6          ; POURSUITE DU CYCLE SHELL
EB                 EX      DE,HL
DDE5               PUSH    IX
FDE1               POP     IY              ; RESTAURE PTR DE E$(J+H)
DDE5               PUSH    IX
E1                 POP     HL
ED42               SBC     HL,BC
ED42               SBC     HL,BC
ED42               SBC     HL,BC
E5                 PUSH    HL
DDE1               POP     IX              ; RESTAURE PTR DE E$(J)
EB                 EX      DE,HL
1899               JR      TRI3            ; RETOUR ARRIERE (METZNER)
                   END
0000
00000 TOTAL ERRORS
```

Cluedo



Minuit... l'heure du crime...........................
Un cri..... Ha!....... c'est horrible...............
Le Docteur a été assassiné..........................
Qui l'a tué???......
L'inspecteur est sur les lieux.Il va vous aider dans cette enquête.........
En effet, vous aurez à trouver : Le criminel, l'endroit du crime (car le corps a été brulé et transporté),et l'arme du crime.

Ici, l'ordinateur choisit "au hasard" lui-même le criminel, le lieu et l'arme du crime. Puis, il donne 3 faits réels après vous avoir donné la liste des possibilités.

Maintenant, à vous de jouer...
Pour commencer, faire SHFT =.
Vous voyez apparaitre : RND =
Vous devez rentrer un nombre entre 0 et 1, par exemple : RND = 0.270568

Vous devez rentrer un nombre

Après un petit moment le programme affiche la liste des suspects, puis celle des armes, puis celle des lieux.
Puis, les 3 indices.relatifs aux suspects, aux armes et aux lieux.
Et enfin commence le jeu. Vous voyez apparaitre sur votre "écran" :

ORDRE :

Vous avez le choix entre :

S pour voir la liste des suspects
A pour voir la liste des armes
L pour voir la liste des lieux
IN pour avoir un indice (qui coûte 0,5 point sur 10 au au départ).
L'indice est pris au hasard entre les suspects, les armes et les lieux. Un indice peut donc se répéter (il faut avoir de la chance...) Mais l'information est toujours vraie.
-SOLU pour pouvoir rentrer son hypothèse. Si elle est fausse, celle-ci est démentie ; mais si elle est vraie, alors elle est confirmée.
Supposons que vous tapiez :

ORDRE : IN.

Vous obtiendrez alors par exemple :
"La cuisine était fermée... au moment du crime".
(Ceci est un exemple de ce qui est choisi au hasard vous pouvez aussi voir affiché un autre texte).
Puis s'affiche à nouveau ORDRE :
Maintenant, tapons

ORDRE : IN.

Nous obtiendrons par exemple "Il n'y avait pas de COLT... au moment du crime".
Faisons une nouvelle demande :

ORDRE : IN.

pas de chance... nous revoyons s'afficher : "Il n'y avait pas de COLT au moment du crime".

Essayons à nouveau :

ORDRE : IN.

"La cuisine était fermée... au moment du crime..."

Continuez à lui demander des indices jusqu'à ce que vous pensiez connaitre les trois vérités.

Mais attention: chaque fois que vous demandez un indice, le programme va vous retirer 0.5 point sur votre score total: on arrive vite à un mauvais score!

Vous pouvez entrer :

ORDRE : SOLU

Le programme vous demande :

CRIMINEL ?

Vous devez rentrer le nom du criminel auquel vous pensez, par exemple :

CRIMINEL ? LEBLANC

Ensuite, vous voyez apparaitre :

ARME ?

Vous devez rentrer l'arme du crime à laquelle vous pensez :

ARME ? COUTEAU

De même pour les lieux :

LIEU ? VERANDA

Alors, si ce ne sont pas les trois vérités, l'O.P. vous retire 3 points sur votre score.

Si par contre, vous avez trouvé par exemple le bon criminel, il ne vous retire que deux points sur votre score.

Et si par exemple, vous avez trouvé le criminel, et le lieu, il ne vous retire qu'un point sur votre score.

Enfin si vous avez trouvé le criminel, le lieu et l'arme du crime, alors (et seulement alors), il ne vous retire rien, mais vous affiche votre score. On arrive vite au score négatif, n'est-ce pas?

Lorsque vous faite une erreur (ou deux), mais que vous avez trouvé, par exemple, l'arme du crime, le programme ne vous demande à la prochaine entrée de SOLU que les rubriques où vous avez fait des erreurs: si vous l'avez trouvée, il ne vous demandera plus l'arme du crime.

NOTATION :

10	: tricheur...
9,5	: où avez-vous volé votre trèfle à quatre feuilles ?!
9	: très bien
8,5	: bien
8	: pas mal
7,5	: ça peut aller
7	: moyen
6,5	: c'est juste
6	: attention, ça a craqué
5,5	:ça y est, ça a craqué
5	: presque nul
4,5	: nul
4	: moins que nul
3,5 à 1	: attention, on va tomber
0 à - 5	: Ouh !.... Ouh !... allez vous cacher...
moins de - 5	: ça y est : le pocket a honte d'avoir un maître aussi nul au CLUEDO : il va s'auto détruire !!...

Après cette description du jeu qui, on peut le dire, est assez différent du "vrai" CLUEDO, il ne vous reste plus maintenant qu'à rentrer le programme. Il es long et fastidieux à copier, et attention : tapez le exactement comme il est imprimé ici.

```
  4:"R"A=A+π:A=A
    AAAA:A=A-INT
    A:N=C+INT (D
    A):RETURN
  5:"S"BEEP 1:
    PAUSE "VOICI
     LES SUSPECT
    S :":FOR E=1
    5TO 19:PRINT
    "M. ";A$(E):
    NEXT E:
    RETURN
  6:"A"BEEP 1:
    PAUSE "VOICI
     LES ARMES :
    ":FOR E=20TO
    25:PRINT "LE
     ";A$(E):
    NEXT E:
    RETURN
  7:"L"BEEP 1:
    PAUSE "VOICI
     LES LIEUX :
    ":FOR E=26TO
    31:PRINT "LA
     ";A$(E):
    NEXT E:
    RETURN
 10:"="H=10:
    INPUT "RND="
    ;A:I$="":K$=
    "":L$="
 20:C=15:D=5:
    GOSUB "R":B$
    =A$(N):C=20:
    D=6:GOSUB "R
    ":F$=A$(N):C
    =26:D=6:
    GOSUB "R":G$
    =A$(N)
 30:GOSUB "S":
    GOSUB "A":
    GOSUB "L
 40:BEEP 3:PAUSE
    "VOUS SAVEZ
    QUE....":C=1
    5:D=5:GOSUB
    "R
 50:IF A$(N)=B$
    GOSUB "R":
    GOTO 50
 55:BEEP 2:PRINT
    "M. ";A$(N);
    " N^A PAS TU
    E...":PAUSE
    "LE DOCTEUR
    LENOIR...
 60:C=20:D=6:
    GOSUB "R
 70:IF A$(N)=F$
    GOSUB "R":
    GOTO 70
 75:BEEP 2:PAUSE
    "QU^ON NE L^
    A PAS TUE...
    ":PRINT "AVE
    C UN ";A$(N)
    ;"...
 80:C=26:D=6:
    GOSUB "R
 90:IF A$(N)=G$
    GOSUB "R":
    GOTO 90
 95:BEEP 2:PAUSE
    "ET QU^IL N^
    EST PAS MORT
    ":PRINT "DAN
    S LA ";A$(N)
    ;".
100:C=15:D=16:
    GOTO 150
110:"IN."H=H-.5:
    GOSUB "R
115:IF N>19THEN
    135
120:IF A$(N)=B$
    GOSUB "R":
    GOTO 115
130:BEEP 1:PRINT
    "M. ";A$(N);
    " ETAIT MALA
    DE":GOTO 145
135:IF N>25THEN
    140
136:IF A$(N)=F$
    GOSUB "R":
    GOTO 115
137:BEEP 1:PAUSE
    "IL N^Y AVAI
    T PAS...":
```

```
    PRINT "DE ";
    A$(N);"...":
    GOTO 145
140:IF A$(N)=G$
    GOSUB "R":
    GOTO 115
141:BEEP 1:PRINT
    "LA ";A$(N);
    " ETAIT FERM
    EE
145:PAUSE "AU MO
    MENT DU CRIM
    E...":RETURN
150:M=0:INPUT "O
    RDRE :";J$:
    GOSUB J$
155:GOTO 150
190:"SOLU."IF I$
    =B$THEN 215
195:BEEP 1:INPUT
    "CRIMINEL ?"
    ;I$
200:IF B$=I$BEEP
    1:PRINT "M.
    ";B$;" EST B
    IEN L^AS":
    PRINT "-SASS
    IN DU DC LEN
    OIR.":GOTO 2
    15
210:M=M+1
215:IF K$=F$THEN
    240
220:BEEP 1:INPUT
    "ARME ?";K$:
    IF K$=F$BEEP
    1:PRINT "LE
    ";F$;" EST B
    IEN L^AR":
    PRINT "-ME D
    U CRIME...":
    GOTO 240
230:M=M+1
240:IF L$=G$GOTO
    255
250:BEEP 1:INPUT
    "LIEU ?";L$:
    IF G$=L$BEEP
    1:PRINT "LA
    ";G$;" EST B
    IEN...":
    PRINT "L^END
    ROIT DU CRIM
    E.
255:M=M+1:IF G$=
    L$M=M-1:IF M
    =0PRINT "SCO
    RE=";H:END
260:H=H-M:BEEP M
    :PAUSE "CE N
    E SONT PAS..
    .":PRINT "LE
    S 3 VERITES.
    ":RETURN
```

Organisation de la mémoire :

A	: le nombre entre 0 et 1
A $ (15 - 31)	: suspects, armes, lieux
B	: le criminel
C, D,	: borne inferieure, et borne supérieure
E	: boucles FOR... NEKT
F	: l'arme
G	: le lieu
H	: le score
I	: le criminel (hypothèse du joueur)
J	: Ordre
K	: Arme (hypothèse du joueur)
L	: Lieu (hypothèse du joueur)
M	: points retirés au score
N	: le RND calculé à partir de A

Pour jouer, il faut, après avoir rentré le programme, rentrer les datas correspondantes
En voici la liste :

```
A$(15.)=VIOLET     A$(23.)=POISON
A$(16.)=LEBLANC    A$(24.)=MARTEAU
A$(17.)=OLIVE      A$(25.)=COLT
A$(18.)=ROSE       A$(26.)=BIBLIO.
A$(19.)=MOUTARD    A$(27.)=CUISINE
A$(20.)=COUTEAU    A$(28.)=CHAMBRE
A$(21.)=LASSO      A$(29.)=VERANDA
A$(22.)=RASOIR     A$(30.)=CAVE
                   A$(31.)=COUR
```

LA TABLE DE JEU

Il s'agit d'un petit programme qui permet de mieux jouer au CLUEDO. Pour ce faire, vous devez utiliser votre imprimante.
Celle-ci imprime une table de jeu pour que vous nayez plus qu'à cocher les suspects (armes ou lieux) que vous innocentez.
Après avoir rentré le programme, allumez votre imprimante et faites un mode RUN, Run.

```
  ?? CLUEDO ??
+---SUSPECTS---+
+ VIOLET  +-  -+
+ LEBLANC +-  -+
+ OLIVE   +-  -+
+ ROSE    +-  -+
+ MOUTARD +-  -+
+-----ARMES----+
+ COUTEAU +-  -+
+ LASSO   +-  -+
+ RASOIR  +-  -+
+ POISON  +-  -+
+ MARTEAU +-  -+
+ COLT    +-  -+
+-----LIEUX----+
+ BIBLIO. +-  -+
+ CUISINE +-  -+
+ CHAMBRE +-  -+
+ VERANDA +-  -+
+ CAVE    +-  -+
+ COUR    +-  -+
+--------------+
```

```
 5:PRINT "  ??
   CLUEDO ??
10:PRINT "+---S
   USPECTS---+
20:PRINT "+ VIO
   LET  +-  -+"
   :PRINT "+ LE
   BLANC +-  -+
   ":PRINT "+ O
   LIVE   +-  -
   +"
25:PRINT "+ ROS
   E    +-  -+"
   :PRINT "+ MO
   UTARD +-  -+
   ":PRINT "+--
   ---ARMES----
   +"
26:PRINT "+ COU
   TEAU +-  -+"
   :PRINT "+ LA
   SSO   +-  -+
   ":PRINT "+ R
   ASOIR  +-  -
   +"
27:PRINT "+ POI
   SON  +-  -+"
   :PRINT "+ MA
   RTEAU +-  -+
   ":PRINT "+ C
   OLT    +-  -
   +"
28:PRINT "+----
   -LIEUX----+"
   :PRINT "+ BI
   BLIO. +-  -+
   ":PRINT "+ C
   UISINE +-  -
   +"
29:PRINT "+ CHA
   MBRE +-  -+"
   :PRINT "+ VE
   RANDA +-  -+
   ":PRINT "+ C
   AVE    +-  -
   +"
30:PRINT "+ COU
   R    +-  -+
35:PRINT "+----
   ----------+
```

Il peut être amusant de faire le montage suivant :
- copier le programme de table de jeu
- rajouter
  40 : INPUT ≠"CLUEDO", A $ (15)
  50 : CHAIN "CLUEDO", "="
- enregistrer le programme

- copier, puis enregister les datas en faisant
  PRINT ≠ "CLUEDO"; A$(15)
- copier, puis enregistrer le programme en faisant :
  CSAUE "CLUEDO"

Lorsque vous aurez envie de jouer, vous n'aurez plus qu'à faire : LLOAD "nom du programme de table de jeu".

Ensuite, allumez l'imprimente et faites RUN. Allez boire une tasse de café et revenez 2 minuites après : vous aurez une table de jeu imprimée, et vous serez prêt à jouer.

Si vous ne possedez pas d'imprimante, photocopiez La table de jeu donnée ici et découpez-là selon vos besoins.

Une dernière chose : ne jouez pas à ce jeu dans le métro, vous risqueriez de rater votre station.
Si, par hasard, il y a une interruption de jeu (ou si vous avez appuyé sur le bouton OFF) ne tuez pas quelqu'un... rallumez votre ordinateur et faite : GOTO 150 (en effet, il est désagréable d'interrompre une partie de CLUEDO, surtout si la chance vous sourit...)

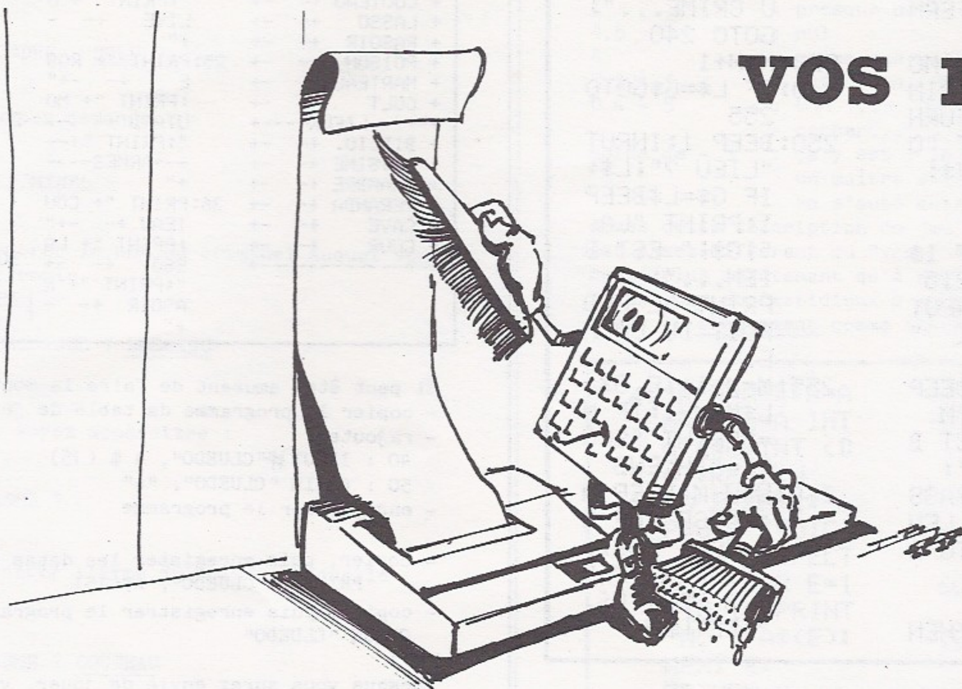
Maintenant, je vous laisse en espérant que vous vous détendrez tranquillement : Bonne chance !

Jean-David BLANC

Analyse du programme

ligne 4	: formule classique du RND
ligne 5	: liste des suspects
ligne 6	: liste des armes
ligne 7	: liste des lieux
ligne 10	: début du jeu initialisation du jeu
ligne 20	: choix des suspects, armes, lieux
ligne 30	: énumération des suspects, armes, lieux
ligne 40-100	: premiers indices "gratuits"
ligne 110-150	: indices payants
ligne 150	: ordre
ligne 190-255	: solution
ligne 260	: affichage du score

Un programme pour tapisser vos murs



Ce programme sans prétention a pour but d'aider les bricoleurs du dimanche à calculer le nombre de rouleaux de papier peint ou le métrage de tissu nécessaires pour tapisser une pièce.

Les données sont les suivantes :

M : nombre de murs à recouvrir (cas le plus courant 4)
H : hauteur entre plafond et plinthe (ex.:2,35 m)
T : périmètre correspondant aux murs à recouvrir (ex.: 20 m)
P : somme des largeurs des portes et portes-fenêtres (ex.: 2 portes = 1,60 m)
F : somme des largeurs des fenêtres (ex.: 2 fenêtres = 2,40 m)
Q : hauteur maximale au-dessus des portes, fenêtres et portes-fenêtres (ex.: 0,50 m correspondant aux portes)
E : hauteur maximale au-dessous des fenêtres (ex.:1 m)
W : largeur du rouleau de papier peint (ex.: 0,55 m) ou du tissu (ex.: 2,70 m)
L : longueur du rouleau de papier peint (ex.: 10,50 m)
R : hauteur du raccord éventuel (ex.: 0,20 m)
V : largeur de la lisière du tissu (ex.: 0,02 m)

La méthode de calcul est relativement simple et diffère légèrement suivant que l'on pose du papier, du tissu en grande largeur ou du tissu en petite largeur.

° Dans le cas du papier peint on calcule successivement les valeurs suivantes :

A : nombre maximal de "grands lés" correspondant à la hauteur de la pièce et sans tenir compte des portes, ni des fenêtres, ni des portes-fenêtres
B : nombre minimal de "grands lés" en tenant compte des portes, des fenêtres et des portes-fenêtres
C : nombre de lés "de dessus", correspondant à la hauteur Q
D : nombre de lés "de dessous", correspondant à la hauteur E
N : nombre de "grands lés" dans un rouleau de papier peint
X : nombre maximal de rouleaux
Y : nombre minimal de rouleaux.

Après pose des B "grands lés", sur les Y rouleaux il reste S mètres dans lesquels on peut découper G "lés de dessus" ; si G est inférieur au nombre requis C, il faut un rouleau de plus et les calculs sont terminés ; dans le cas contraire, on essaie de découper G"lés de dessous" dans les S mètres restant ; si G est inférieur au nombre D, il faut un rouleau supplémentaire. On calcule ainsi Y qui en aucun cas ne peut être supérieur à X.

° Dans le cas du tissu en petite largeur on calcule également A et B mais en tenant compte des "lisières" éventuelles, puis C et D. Le métrage maximal X ne tient pas compte des portes, fenêtres et portes-fenêtres ; le métrage minimal est calculé en essayant d'optimiser les découpes.

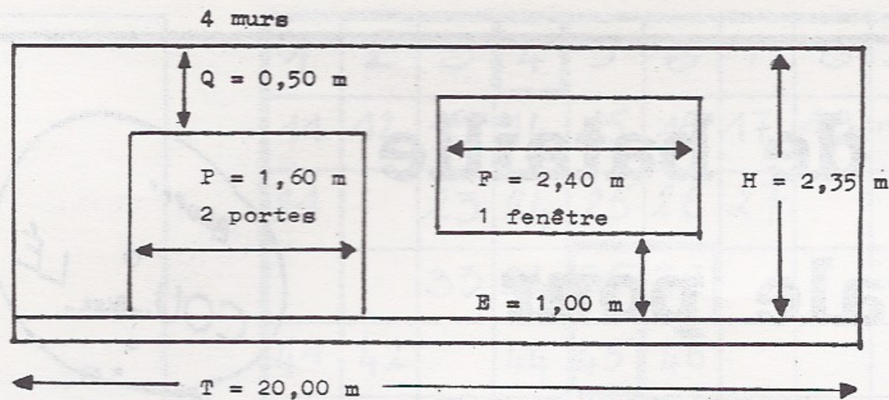
° Le tissu en grande largeur peut être posé verticalement (par exemple s'il y a des motifs...) ; on est alors ramené au cas précédent. S'il est posé horizontalement, les calculs sont encore plus simples ; le métrage maximal ne tient compte que du périmètre et du nombre de panneaux ; le métrage minimal n'a de sens que si la largeur cumulée des portes, fenêtres et portes-fenêtres est inférieure à la largeur du tissu.

J'ai oublié,... toutes les valeurs numériques sont en mètres. Et maintenant le plus dur reste à faire, coller le papier ou poser le tissu ... BON COURAGE !

Alain GANDOSSI



PIECE 1

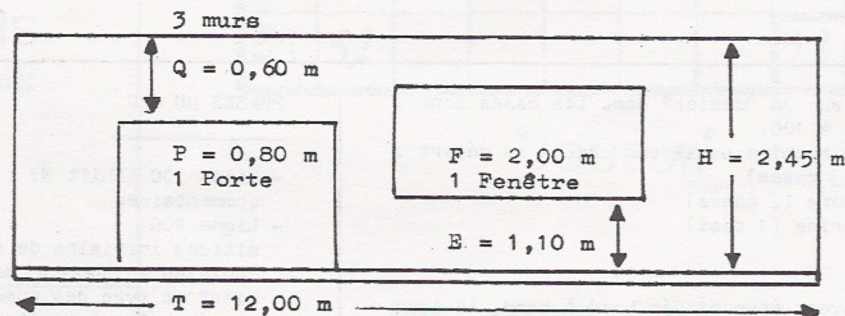


Papier L = 10,50 m W = 0,55 m R = 0,10 m

Tissu 1 W = 1,40 m V = 0,02 m R = 0,20 m

Tissu 2 W = 2,70 m V = 0,02 m R = 0

PIECE 2



Papier L = 11,00 m W = 0,55 m R = 0,05 m

Listing

```
  1:REM *TAPISS
  2:REM *AUTEURS
    :ALAIN^ET^JE
    ROME^GANDOSS
    I
  3:REM *COPYRIG
    HT:TRACE^ET^
    LES^AUTEURS
  4:PAUSE "D:DEB
    UT"
  5:PAUSE "N:NOU
    VEL ESSAI"
  9:END
100:"D"PAUSE "PI
    ECE...":
    PAUSE "DIMEN
    SIONS EN MET
    RES..."
105:INPUT "NB MU
    RS?";M
110:INPUT "HAUT
    MAX PLAF-PLI
    NTHE?";H
115:INPUT "PERIM
    ETRE?";T
120:INPUT "SOMME
     LARG PORT/P
    -FEN?";P
125:INPUT "SOMME
    LARG FENETR
    ES?";F
130:IF T<=P+F
    GOSUB 800:
    GOTO 120
140:INPUT "HAUT
    MAX DESSUS P
    /F?";Q
145:INPUT "HAUT
    MAX DESSOUS
    FEN?";E
150:IF H<=Q+E
    GOSUB 800:
    GOTO 140
160:PAUSE "QUE P
    OSEZ-VOUS?":
    INPUT "PAPIE
    R(1)/TISSU(2
    )?";G
165:IF (G<1)+(G>
    2)THEN 160
170:GOTO E2+E2*G
175:"N"INPUT "NO
    UVEL ESSAI?"
    ;G$:IF G$="N
    "PRINT "AU R
    EVOIR...":
    END
180:IF G$="O"
    GOTO 190
185:GOTO "N"
190:INPUT "MEME
    PIECE?";G$:
    IF G$="O"
    GOTO 160
195:IF G$="N"
    GOTO E2
196:GOTO 190
200:PAUSE "PAPIE
    R PEINT..."
205:INPUT "LONG
    ROULEAU?";L
210:INPUT "LARG
    ROULEAU?";W
215:INPUT "HAUT
    RACCORD?";R:
    IF R>LGOSUB
    800:GOTO 215
220:GOSUB 700:N=
    INT (L/(H+R)
    ):X=A/N:IF X
    <>INT XLET X
    =1+INT X
225:Y=B/N:IF Y<>
    INT YLET Y=1
    +INT Y
230:S=YL-B*(H+R)
    :G=INT (S/(Q
    +R)):IF G<C
    GOTO 245
235:S=S-C*(Q+R):
    G=INT (S/(E+
    R)):IF G<D
    THEN 245
240:S=S-D*(E+R):
    GOTO 250
245:Y=Y+1
250:IF Y>XLET Y=
    X
255:PRINT "NB RO
    ULEAUX:"
260:PRINT USING
    "###";"MAX="
    ;X:PRINT "MI
    N=";Y:GOTO "
    N"
300:PAUSE "TISSU
    ..."
305:INPUT "LARG
    TISSU?";W
310:INPUT "LARG
    LISIERE?";V
315:INPUT "HAUT
    RACCORD?";R
320:IF W-.1>H
    GOTO 500
400:PAUSE "PETIT
    E LARGEUR!":
    PAUSE "FAIRE
     DES COUTURE
    S!"
405:W=W-2V:GOSUB
    700:X=A*(H+R
    +.1)+.1M
410:Y=B*(H+R+.1)
    +.1M+C*(Q+R+
    .1)+D*(E+R+.
    1):IF Y>XLET
    Y=X
415:PRINT "METRA
    GE:"
420:PRINT USING
    ;"MAX=";X:
    PRINT "MIN="
    ;Y:GOTO "N"
500:PAUSE "GRAND
    E LARGEUR!"
505:INPUT "POSE
    HORIZ(H)/VER
    T(V)?";G$:IF
    G$="V"GOTO 4
    05
510:IF G$="H"
    GOTO 520
515:GOTO 505
520:X=T+.1M:IF P
    +F>W-.1LET Y
    =X
525:Y=T-P-F+.1M+
    E+Q:GOTO 415
700:A=T/W:IF A<>
    INT ALET A=1
    +INT A
705:B=(T-P-F)/W:
    IF B<>INT B
    LET B=1+INT
    B
710:C=(P+F)/W:IF
    C<>INT CLET
    C=1+INT C
715:D=F/W:IF D<>
    INT DLET D=1
    +INT D
720:RETURN
800:PAUSE "ERREU
    R!":RETURN
```

Exemples

```
NB ROULEAUX:
MAX= 10
MIN=  8

METRAGE:
MAX=40.15
MIN=37.2

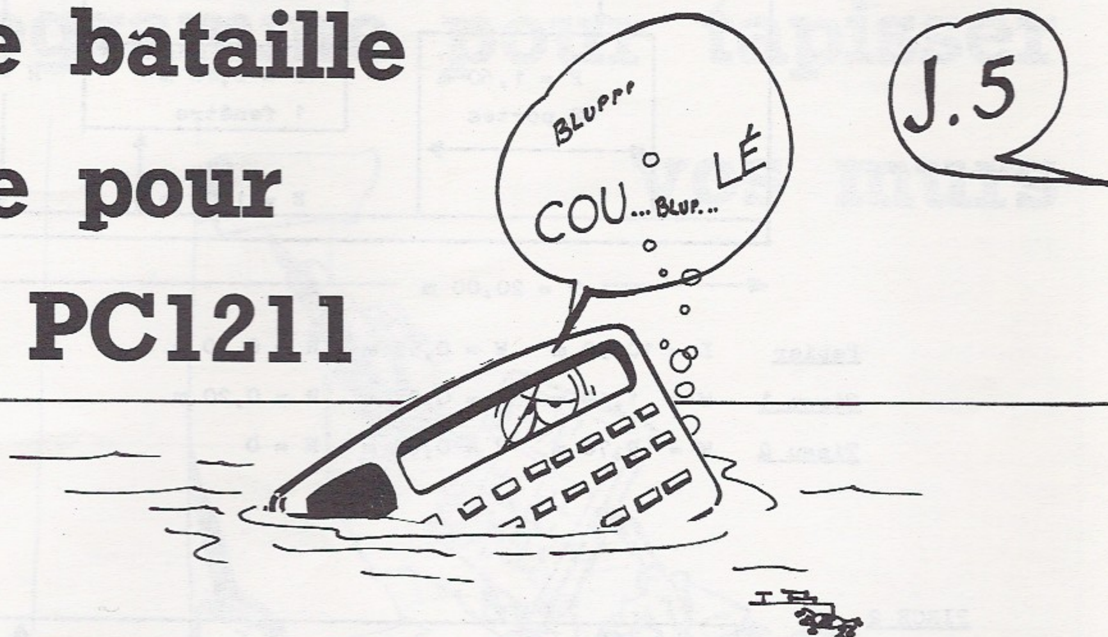
METRAGE:
MAX=20.4
MIN=17.9

METRAGE:
MAX=20.
MIN=19.85

NB ROULEAUX:
MAX=  6
MIN=  5

AU REVOIR...
```

Jeu de bataille navale pour Sharp PC1211



Le jeu se joue sur un "damier" dont les cases sont numérotées de 1 à 100.
Le Joueur et la Machine possèdent chacun au départ :
- un Cuirassé (3 cases)
- deux Torpilleurs (2 cases)
- trois Sous-Marins (1 case)

Les navires peuvent être placés bord à bord, le long des côtés du damier ou peuvent se toucher par les coins ; par contre les éléments les constituant doivent être alignés ou placés dans des directions perpendiculaires (voir figure).

Le programme ne mémorise que les éléments correspondant au jeu de la machine, le joueur devant tenir à jour ses propres tableaux.

Les mémoires A B ...J contiennent les numéros des dix cases correspondant aux six navires de la Machine ; chaque fois que l'un d'entre eux est "touché", la valeur 1000 est ajoutée à la mémoire correspondante ; quand le navire est "coulé", elle est remise à zéro.

Le nombre de pièces de la Machine touchées ou coulées est rangé en A(27).
La Machine range dans les mémoires K L ...T les positions des pièces de l'adversaire (le Joueur) qu'elle a trouvé, c'est-à-dire à chaque annonce d'un "touché" ou "coulé". Elle utilise les mémoires A(28) à A(31) pour ranger divers éléments lui permettant d'optimiser son jeu :

- A(28) : Pointeur de la liste des pièces de l'adversaire touchées ou coulées
- A(29) : Mémorisation de la dernière position jouée
- A(30) : Mémorisation d'une position antérieure
- A(31) : Constante utilisée pour l'exploration du damier

Les mémoires U V ...Z sont utilisées pour ranger diverses variables de travail.

PHASES DU JEU

- Ligne 100 (Shift N) : Remise à zéro générale et commentaires
- Ligne 200 : Choix par la Machine des positions initiales de ses navires. Des valeurs sont choisies au hasard, mais dans les limites du "damier" et avec des numéros "adjacents" pour le cuirassé et les torpilleurs.
  Choix du premier joueur au hasard.
- Ligne 300 (Shift M) : La machine joue ("à moi").
  Au départ elle choisit une case au hasard et pose une question au joueur :
  ° si la réponse est 0 (à l'eau!), elle lui passe la main
  ° si la réponse est 1 (touché!) ou 2 (coulé), elle mémorise la position, déplace le pointeur de pile A(28), mémorise le coup en A(29) et passe la main; si A(28) atteint la valeur 20 elle a gagné.
  Le coup suivant elle explore le jeu en utilisant un algorithme simple en vérifiant que la case n'a pas déjà été touchée ou coulée.
  Si la réponse est 1, elle explore systématiquement les cases adjacentes de A(29) à chaque "à l'eau" suivant ; si cette recherche ne donne rien, elle revient au point précédent A(30) et fait la même chose en évitant les positions déjà touchées.
  Si la réponse est 2, elle reprend sa recherche à partir du point précédent A(30).
- Ligne 400 (Shift M) : Le Joueur joue ("à vous").
  La machine demande un numéro de case et vérifie qu'il ne sort pas des limites ; elle effectue une comparaison avec ses positions mémorisées en A B ...J et répond en conséquence ; elle pose des "drapeaux" dans le cas où le navire a été "touché" ou "coulé" et mémorise en A(27) le nombre de pièces dans ce cas.
  Si A(27) est égal à 10, elle a perdu.

SOUS-PROGRAMMES UTILISES

- Ligne 2 : donne une valeur aléatoire X comprise entre 1 et 100.
- Ligne 3 : recoit une position X
  donne une position adjacente Y=X+UV avec U= +1 ou -1 et V= 1 ou 10

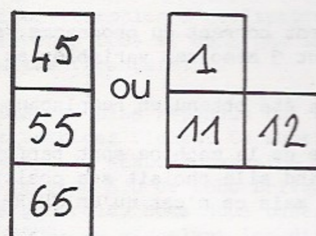
PIECES

1 CUIRASSE
= 3 CASES

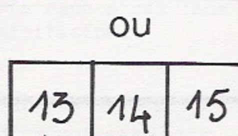
2 TORPILLEURS
= 2 CASES

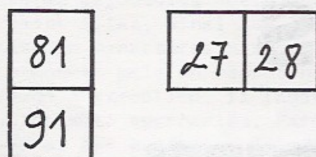
3 SOUS-MARINS
= 1 CASE

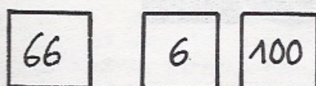
EXEMPLES



ou







1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21		23	24	25	26	27			
		33	34	35	36				
41	42		44	45	46				
51	52								
	62	63							
	72								
	82								
91	92						98	99	100

"JOUEUR"

"MACHINE"

1	2	3	4	5	6	7	8	9	10
			14	15	16				
		23							
		33							
		43							
51	52	53							
	62								
	72			75	76				
	82			85					
91	92								

POQUETTE

LISTING

```
  1:REM "BATNAV
    COPYRIGHT  :
    A.GANDOSSI
    ET TRACE
  2:GOSUB 9:X=
    INT (E2W+1):
    RETURN
  3:GOSUB 9:U=-3
    +2*INT (2W+1
    ):GOSUB 9:V=
    INT (2W+1):V
    =INT (V^3.4)
    :Y=UV:GOSUB
    20:IF U=1
    GOTO 3
  4:Y=Y+X:RETURN
  7:U=0:FOR Y=1
    TO V-1:IF A(
    V)=A(Y)LET U
    =U+1
  8:NEXT Y:
    RETURN
  9:W=W+π:W=WWWW
    W:W=W-INT W:
    RETURN
 20:U=1:IF Y=-1
    IF X-10*INT
    (X/10)=1
    RETURN
 21:IF Y=1IF X/1
    0=INT (X/10)
    RETURN
 22:IF Y+X<=0
    RETURN
 23:IF Y+X>E2
    RETURN
 24:U=0:RETURN
 30:A=0:B=0:C=0
 35:BEEP 3:PAUSE
    "COULE!":
    RETURN
 40:X=A(30)+A(31
    ):IF X>E2LET
    X=X-E2
 41:A(30)=X:
    RETURN
 50:IF V=5LET V=
    1:A(29)=A(30
    ):RETURN
 51:Y=(10^V)/10:
    IF Y>10LET Y
    =-Y/E2
 52:V=V+1:GOSUB
    20:IF U=1
    GOTO 50
 53:X=X+Y:RETURN
100:"N"CLEAR :
    BEEP 5:INPUT
    "NB SVP?";W:
    W=ABS SIN W
120:PRINT "NAVIR
    ES:1*3 2*2 3
    *1
200:BEEP 3:PAUSE
    "JE CHOISIS
210:GOSUB 2:A=X:
    GOSUB 3:B=Y
215:X=B:GOSUB 3:
    C=Y:IF C=A
    GOTO 215
220:FOR Z=1TO 2:
    BEEP 1
225:GOSUB 2:V=2Z
    +2:A(V)=X:
    GOSUB 7:IF U
    <>0GOTO 225
230:GOSUB 3:V=2Z
    +3:A(V)=Y:
    GOSUB 7:IF U
    <>0GOTO 230
235:NEXT Z
240:FOR Z=8TO 10
    :BEEP 1
245:GOSUB 2:A(Z)
    =X:V=Z:GOSUB
    7:IF U<>0
    GOTO 245
250:NEXT Z:A(28)
    =11:GOSUB 2:
    A(30)=X:A(31
    )=53:V=1
255:BEEP 5:PRINT
    "J^AI FINI E
    T VOUS?
260:PAUSE "CHOIX
     JOUEUR 1":
    GOSUB 2:IF X
    >50GOTO 400
300:"M"BEEP 1:
    PAUSE "A MOI
    !
301:IF A(29)=0
    GOSUB 40:
    GOTO 303
302:X=A(29):
    GOSUB 50
303:U=0:FOR Z=11
    TO A(28):IF
    X=A(Z)LET U=
    1
304:NEXT Z:IF U=
    1GOTO 301
305:BEEP 1:PAUSE
    "JE JOUE EN
    ";X:INPUT "V
    OTRE REPONSE
    (0/1/2)?";U:
    IF (U<0)+(U>
    2)GOTO 305
310:IF U=0GOTO 4
    00
315:A(A(28))=X:
    IF A(28)=20
    BEEP 5:PRINT
    "J^AI GAGNE!
    ...":END
320:A(28)=A(28)+
    1:V=1:IF U=2
    LET A(29)=0:
    GOTO 400
325:A(29)=X
400:"V"BEEP 1:
    PAUSE "A VOU
    S!":INPUT "N
    O CASE?";X:
    IF (X<=0)+(X
    >E2)GOTO 400
405:U=0:FOR Z=1
    TO 10:IF X<>
    A(Z)GOTO 435
410:U=1:IF (Z=8)
    +(Z=9)+(Z=10
    )GOSUB 35:A(
    Z)=0:GOTO 43
    5
415:IF (Z=4)*(E>
    E3)+(Z=5)*(D
    >E3)GOSUB 35
    :D=0:E=0:
    GOTO 435
420:IF (Z=6)*(G>
    E3)+(Z=7)*(F
    >E3)GOSUB 35
    :F=0:G=0:
    GOTO 435
425:IF (Z=1)*(B>
    E3)*(C>E3)+(
    Z=2)*(A>E3)*
    (C>E3)+(Z=3)
    *(A>E3)*(B>E
    3)GOSUB 30:
    GOTO 435
430:BEEP 2:PAUSE
    "TOUCHE!":A(
    Z)=A(Z)+E3
435:NEXT Z:IF U=
    0BEEP 1:
    PAUSE "A L^E
    AU!":GOTO 30
    0
440:A(27)=A(27)+
    1:IF A(27)=1
    0BEEP 5:
    PRINT "J^AI
    PERDU!...":
    END
450:GOTO 300
```

EXEMPLE

```
NAVIRES:1*3 2*2
3*1
JE CHOISIS
J^AI FINI ET VOU
S?
CHOIX JOUEUR 1
A VOUS!
              37.
A L^EAU!
A MOI!
JE JOUE EN 92.
               0.
A VOUS!
              56.
A L^EAU!
A MOI!
JE JOUE EN 45.
               1.
A VOUS!
              16.
COULE!
A MOI!
JE JOUE EN 46.
               0.
A VOUS!
               5.
A L^EAU!
A MOI!
JE JOUE EN 55.
               0.
A VOUS!
              23.
TOUCHE!
A MOI!
JE JOUE EN 44.
               0.
A VOUS!
              76.
COULE!
A MOI!
JE JOUE EN 35.
               1.
A VOUS!
              38.
A L^EAU!
A MOI!
JE JOUE EN 36.
               2.
A VOUS!
               0.
A VOUS!
             101.
A VOUS!
              33.
TOUCHE!
A MOI!
JE JOUE EN 98.
               1.
A VOUS!
              43.
COULE!
A MOI!
JE JOUE EN 99.
               2.
A VOUS!
              46.
A L^EAU!
A MOI!
JE JOUE EN 51.
               1.
A VOUS!
               1.
A L^EAU!
A MOI!
JE JOUE EN 52.
               0.
A VOUS!
               7.
A L^EAU!
A MOI!
JE JOUE EN 61.
               0.
A VOUS!
              51.
COULE!
A MOI!
JE JOUE EN 41.
               2.
A VOUS!
              10.
A L^EAU!
A MOI!
JE JOUE EN 4.
               2.
A VOUS!
             100.
A L^EAU!
A MOI!
JE JOUE EN 57.
               6.
JE JOUE EN 57.
              -1.
JE JOUE EN 57.
               0.
A VOUS!
              75.
TOUCHE!
A MOI!
JE JOUE EN 10.
               2.
A VOUS!
               9.
A L^EAU!
A MOI!
JE JOUE EN 63.
               2.
J^AI GAGNE!...
```

- Ligne 7 : teste si la position A(V) est différente des précédentes
- Ligne 9 : donne une variable aléatoire W < 1 (initialisation à la ligne 100)
- Ligne 20 : teste si une valeur ne sort pas du "cadre"
- Ligne 30 : remise à zéro des positions du cuirassé de la machine et message
- Ligne 35 : imprime un message
- Ligne 40 : donne une valeur X différente de la précédente à chaque appel
- Ligne 50 : recherche des positions adjacentes de X.

COMMENTAIRES

- Pour un fonctionnement correct du programme, supprimer la ligne 1 (il faut 5 mémoires variables en plus des mémoires fixes).
- L'exemple ci-après a été obtenu en remplacant les PAUSE par des PRINT.
- Les temps de réponse de la machine sont parfois un peu longs (surtout quand elle choisit ses positions ou prépare son coup), mais ce n'est qu'un SHARP PC1211.

Alain GANDOSSI

le DOS de A à Z

Chain

Chnon

Clear

Le Modèle 1 possède plusieurs Systèmes d'Exploitation des Disquettes (SED, ou DOS en anglais). Ces différents systèmes ont beaucoup de points communs, particulièrement ceux de la "lignée" TRSDOS-NEWDOS-NEWDOS80-LDOS, qui s'inspirent tous, à quelques détails près, du TRSDOS de Tandy (version initiale 2.1). Nous publierons régulièrement sous forme de fiches pratiques, les commandes de ces différents DOS, classées alphabétiquement. Ces fiches ne prétendent pas remplacer les manuels originaux qu'il est impératif de posséder (il n'était pas envisageable de publier sous cette forme l'équivalent de plusieurs centaines de pages ...).

Leur but est de fournir à l'utilisateur une information rapide d'accès (classement alphabétique), pratique (exemples d'utilisation), comparative (différents DOS) et dans une langue qu'il est censé connaître (français)...

Les DOS les plus répandus ont été retenus pour constituer ces fiches. Ce sont le TRSDOS version 2.3 de Tandy, le NEWDOS80 version 2 d'Apparat et le LDOS version 5.0 de Lobo Drives International. En règle générale, nous nous baserons sur les commandes du TRSDOS, en signalant les différences ou les apports rencontrés dans les autres systèmes d'exploitation.

Conventions d'écriture

Dans la description des commandes, les mots en caractères majuscules désignent les commandes proprement dites, ainsi que les mots-clés associés. Les mots en caractères minuscules ne font pas partie des commandes mais en représentent la forme. Au moment de l'exécution, il faudra leur substituer le ou les mots appropriés. Parmi ceux-ci, nous trouverons par exemple : "nomfich", qui désigne un nom de fichier, "motpasse" qui signifie qu'un mot de passe (password) est attendu à cet endroit. Dans tous les cas, un coup d'oeil sur les exemples donnés devrait lever toute ambiguité.
Enfin, il faut signaler que le NEWDOS80 emploie indifféremment la virgule ou l'espace comme séparateur dans les commandes.

CHAIN nomfich id

Commande NEWDOS 80 uniquement. Voir aussi : DO

But : exécuter automatiquement une suite d'ordres décrits dans un fichier (nomfich) appelé "fichier de chainage. Le mot "DO" (faire, en anglais), peut également être employé. Optionnellement, il est possible de lancer l'exécution à un endroit précis du fichier de chainage, repéré par un identificateur (paramètre id), constitué de 1 à 31 caractères. Cela permet un gain de place appréciable (plusieurs fichiers de commande dans un seul fichier disque). En l'absence de paramètre "id", l'exécution commence au début du fichier.

Le nom du fichier de chainage possède implicitement l'extension "JCL" et, dans ce cas, il n'est pas nécessaire de la mentionner dans la commande CHAIN. Le cas échéant, l'extension doit être précisée.

Le fichier de chainage peut être créé de différentes façons : sous BASIC, par l'éditeur de textes SCRIPSIT ou PENCIL, ou bien encore par le programme utilitaire CHAINBLD/BAS fourni avec le système NEWDOS 80.

Facultativement, ce fichier peut contenir 6 "codes de contrôle" différents, repérés par la suite de caractères "/./" suivie d'un chiffre de 0 à 5, ou par un octet 80H à 85H, et qui ont la signification suivante :

/./0 : indique le nom d'un identificateur de section que l'on pourra invoquer au moment de la commande CHAIN.

/./1 : affichage d'un message avec attente de la frappe du caractère ENTER.

/./2 : le texte qui suit est sauté (sans affichage). Cela permet de placer des commentaires dans le fichier de chainage, sans que ceux-ci soient affichés.

/./3 : affichage d'un message (comme /./1) mais sans attente de ENTER.

/./4 : branchement à l'identificateur de section donne à la suite du code de contrôle.

/./5 : équivaut à la commande CHNON (voir plus loin) Le caractère suivant est obligatoirement : Y, N ou D.

Exemple de fichier de chainage sauvé sous le nom "FICHAIN/JCL" :

```
/./1 DEPART...
DIR
/./3 MESSAGE SANS ATTENTE...
FREE
/./4 FIN
/./OSECTION 2
BASIC
CMD"S"
/./OFIN
/./3 FIN DE CHAINAGE
```

Il y a trois façons d'appeler ce fichier :

```
CHAIN FICHAIN
CHAIN FICHAIN SECTION 2
CHAIN FICHAIN FIN
```

En quoi diffèrent-elles, d'après vous ?

Remarques : Une section se termine par la fin du fichier ou par le début d'une autre section (identificateur). Les identificateurs frappés en caractères minuscules ne semblent pas être digérés correctement !

CHNON paramètre

Paramètre : Y
N
D

But : Cette commande NEWDOS 80 permet l'abandon provisoire, puis la reprise d'exécution du fichier de chainage. Lorsque CHNON,N est rencontré dans le fichier, le chainage est stoppé, le contrôle est rendu au clavier, et toutes les commandes suivantes seront issues de cet organe. Lorsque la commande CHNON,Y est frappée au clavier, le fichier de chainage se poursuit là où il avait été interrompu. Soit, par exemple, le fichier de chainage suivant :

```
DIR
CHNON,N
FREE
```

L'exécution de ce fichier provoque l'affiche du catalogue de la disquette, puis le contrôle est rendu au clavier comme si le chainage était terminé, et on peut ensuite frapper n'importe quelle commande. Lorsque CHNON,Y est entré au clavier, l'exécution du fichier de chainage se poursuit par la commande FREE.

Lorsque la commande CHNON est exécutée à un second niveau d'appel, par exemple dans un programme lancé par le DOS (BASIC par exemple), utilisant la procédure "DOS-CALL" (appel de routine DOS par CMD "commande" sous BASIC), il y a lieu de distinguer deux cas : le chainage se poursuit au niveau du programme appelant (BASIC par exemple), ou le chainage se poursuit au niveau du DOS (appelé). Dans le premier cas, il faudra employer CHNON,Y et dans le second cas : CHNON,D. Nous allons concrétiser cela par le fichier de chainage suivant :

```
DIR
CHNON,N
CMD "S"
FREE
LIB
```

Lorsque le contrôle est rendu au DOS, appelons le BASIC, puis frappons : CMD "CHNON,Y". Le fichier de chainage se poursuit (au niveau BASIC), par CMD "S".

Retirons maintenant la commande CMD "S" du fichier de chainage, et recommencons notre expérience en frappant cette fois sous BASIC : CMD "CHNON,D". Le chainage se poursuit (au niveau DOS) et, à la fin du fichier, le contrôle revient au BASIC. Voilà la différence !

```
CLEAR (paramètres éventuels)

Paramètres :
          START=adr1
          END=adr2
          MEM=adr3
```

But : cette commande NEWDOS 80 est utilisée pour effectuer certaines réinitialisations du DOS. Optionnellement, une zone mémoire peut être remise à zéro.

Les réinitialisations effectuées sont les suivantes :

- routages de périphériques supprimés (voir ROUTE)
- routines accrochées à l'interruption temps-réel désactivées
- routines appelées par "*" désactivées (Spooler)
- réajustement de la zone mémoire protégée (HIMEM) à la valeur donnée par adr3 (valeur maximum si non spécifiée)

Exemple :

CLEAR,START=8000H,END=9000H est remise a zero.
HIMEM est forcée à BFFFH.

Les valeurs implicites prises par défaut sont : 5200H si START n'est pas précisé, HIMEM si END n'est pas précisé :

CLEAR

effectue les réinitialisations ci-dessus et force à zéro la zone comprise entre 5200H et HIMEM.

Index des mots-clés

La liste des mots-clés figurant ci-dessous est destinée à être exploitée par le programme de gestion des articles décrit dans le numéro 1 de TRACE (pages 58 à 64). Le format est le suivant:

Nom abrégé de l'article, numéro de page : 1 à 4 mots-clés.

En prenant l'habitude d'entrer cette liste à chaque numéro de TRACE, vous constituerez ainsi une petite base de données vous permettant de trouver plus efficacement et plus rapidement l'article que vous cherchez. Bien entendu, cette liste est donnée à titre indicatif, et vous pourrez toujours en modifier ou en compléter le contenu, afin de l'adapter à vos besoins ; mais avant toute chose, vous devez vous imposer une certaine normalisation des mots-clés (en principe, ils sont limités à 12 caractères) PROGRAMME par exemple, indiquera toujours qu'un listage de programme est donné dans l'article. Pour cela aussi, faites-nous part de vos idées!

Imprimerie Laballery, 58500 Clamecy. Imprimé en France. Directeur de la publication Jean-Pierre Nizard
Dépôt légal : septembre 1982 - Numéro de commission paritaire : en cours

UN DOCUMENT ESSENTIEL

GUIDE 82-83

DE

L'ORDINATEUR INDIVIDUEL

NOS ESSAIS D'ORDINATEURS

L'ORDINATEUR INDIVIDUEL

GUIDE 82-83



numéro spécial hors série n° 39 bis 30 F

Canada : 5.50 $C - Belgique : 250 FB - Suisse : 10 FS

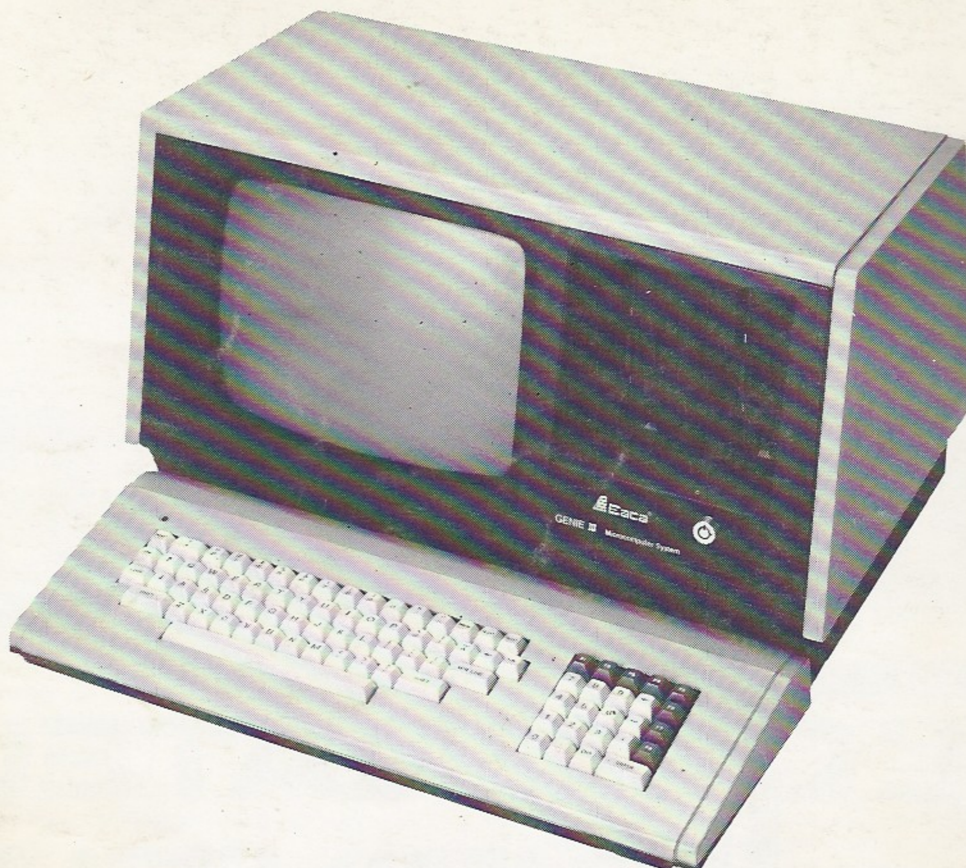
AU SOMMAIRE

- Panorama des ordinateurs valant entre 250 et 60 000 FF (plus de 120 matériels).
- Panorama des imprimantes valant moins de 20 000 FF (plus de 70 matériels).
- Réactualisation de 12 bancs d'essai parus dans l'OI.
- Annuaire des fournisseurs (plus de 700 adresses).
- Annuaire des clubs (plus de 200 adresses).
- Dictionnaire de l'informatique individuelle.
- Le point sur les nouveautés parues depuis l'été 1981.
- Et une série d'articles pour vous "guider" sur le chemin de votre informatisation individuelle.

à partir du 10 septembre 1982
30 FF
chez votre marchand de journaux

Pour recevoir, chez vous le Guide 82-83 dès sa parution, il vous suffit d'envoyer vos nom et adresse ainsi qu'un chèque de 30 FF à L'ORDINATEUR INDIVIDUEL (GUIDE 82-83) 41 rue de la Grange-aux-Belles 75483 Paris Cedex 10

Une réduction de 5 FF est accordée aux abonnés sur envoi de la dernière étiquette d'expédition



Genie III...

Un rapport Prix-Performance inégalé !..

Caractéristiques

- Z80A CPU (3,2 MHz);
- 64 K RAM (extensible 192 K);
- Clavier minuscule/majuscule 85 touches;
- Clavier numérique séparé;
- 8 touches de fonction programmables;
- Ecran vert 12 pouces haute résolution;
- 24 lignes de 80 caractères ou 16 lignes de 64 caractères;
- 2 disquettes 325 K chacune (extensible à 4 de 650 K);
- CPM 2.2 ou NEWDOS 80 version 2;
- Interface parallèle pour imprimante;
- Interface série (modem/communication);
- Horloge en temps réel.
- OPTION graphisme haute résolution.

Professionnel

De part ses performances, il s'adresse aux professionnels mais son prix le rend accessible à une plus large clientèle.

Programmes

Disposant de plusieurs «operating systems» (les plus répandus au monde) et grâce à son affichage commutable, il permet de développer ou d'utiliser l'importante bibliothèque de logiciel écrite sous NEWDOS 80 (GENIE I et II, TRS 80 modèle I) ainsi que les standards disponibles sous CPM (langages utilitaires, etc...).

Pour plus amples renseignements:

Nom: ....

Adresse: ....

....

Téléphone: ....



GENERALE
ELECTRONIQUE
SERVICES

68 et 76 avenue Ledru Rollin - 75012 PARIS
Tél. : 345.25.92 – Télex : 215 546F GESPAR

Editepe